DLV11-J

DLV11-J TEST
CVDLAA0

AH-E189A-MC
COPYRIGHT © 1978
FICHE 1 OF 1

JUL 1978
digital
MADE IN USA

```
        .REM %
```

IDENTIFICATION

PRODUCT CODE: AC-E188A-MC

PRODUCT NAME: CVDLAA0 DLV11-J TEST

PRODUCT DATE: MAY 1978

MAINTAINER: DIAGNOSTIC ENGINEERING

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS DOCUMENT.

NO RESPONSIBILITY IS ASSUMED FOR THE USE OR RELIABILITY OF SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL OR ITS AFFILIATED COMPANIES.

COPYRIGHT (C) 1978 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL	PDP	UNIBUS	MASSBUS
DEC	DECUS	DECTAPE	

TABLE OF CONTENTS

1.0 GENERAL PROGRAM INFORMATION.

1.1 PROGRAM PURPOSE (ABSTRACT).
1.2 SYSTEM REQUIREMENTS.
1.3 RELATED DOCUMENTS AND STANDARDS.
1.4 DIAGNOSTIC HIERARCHY PREREQUISTES.
1.5 ASSUMPTIONS.

2.0 OPERATING INSTRUCTIONS.

2.1 LOADING AND STARTING PROCEDURES.
2.2 SPECIAL ENVIRONMENTS.
2.3 OPERATIONAL SWITCH SETTINGS
2.4 PROGRAM OPTIONS.
2.5 EXECUTION TIMES.
2.6 POWER FAIL.

3.0 ERROR INFORMATION.

3.1 ERROR REPORTING PROCEDURE.
3.2 ERROR HALTS.

4.0 PERFOMANCE AND PROGRESS REPORTS.

4.1 PERFORMANCE REPORTS.

5.0 DEVICE INFORMATION TABLES.

6.0 SUMMARY OF TESTS.

1.0 GENERAL PROGRAM INFORMATION.

1.1 PROGRAM PURPOSE (ABSTRACT).

THIS DIAGNOSTIC IS A LOGIC TEST TO VERIFY THE OPERATION OF THE
DLV11-J SERIAL INTERFACE.
TESTING IS DONE IN TWO DISTINCT PHASES:

1. ALL SELECTED CHANNELS PER DLV11-J MODULE ARE TESTED INDIVIDUALLY

2. THE DLV11-J MODULE IS TESTED AS A WHOLE FOR CHANNEL INTERACTION
   PROBLEMS. THIS DIAGNOSTIC IS DESIGNED TO TEST AND DETECT
   ERRORS TO THE LOGIC LEVEL (NOT TO THE CHIP LEVEL).

THIS DIAGNOSTIC OPERATES UP TO 2 DLV11-J SERIAL LINE INTERFACES
CONFIGURED AT CONSECUTIVE BASE ADDRESSES.
THE PROGRAM WILL DO AUTO-SIZING IF THE DEVICE MAP '$DEVM' = 0.
ALTERNATELY, THE OPERATOR CAN SELECT OR DESELECT INDIVIDUAL MODULES
& CHANNELS BY SETTING THE PROPER BITS IN '$DEVM' (SEE PROGRAM OPTIONS SEC. 2.4)
BEFORE RUNNING THE PROGRAM.
BY THIS METHOD, THE PROGRAM WILL BYPASS AUTO-SIZING AND TEST ONLY
THOSE MODULES/CHANNELS IT FINDS IN $DEVM.

IN EITHER CASE, THE PROGRAM WILL PRINT OUT ALL MODULES & CHANNELS
TO BE TESTED BEFORE PROCEEDING.

THE OPERATOR MUST INSTALL DATA WRAP AROUND CONNECTORS TO DO
DATA TESTING. TO BYPASS DATA TESTS, THE OPERATOR MUST MODIFY
'$USWR' (USER SWITCH REGISTER) , SEE PROGRAM OPTIONS SEC. 2.4)

THE DEFAULT ADDRESSES & VECTORS ARE AS FOLLOWS:

```
        177560 -CONSOLE INTERFACE DEVICE ADDRESS
        176500 -FIRST SERIAL CHANNEL ADDRESS OF UP TO 8  CONSECUTIVE
                SERIAL LINE DEVICES.


           60 - VECTOR FOR CONSOLE DEVICE INTERFACE.
          300 - VECTOR FOR FIRST OF 16 DEVICES.
```

THIS PROGRAM IS DESIGNED TO RUN ON ANY Q-BUS PDP-11 WITH 4K OF
MEMORY AND A DLV11-J (Q-BUS) MODULE. IT CAN RUN UNDER XXDP &
APT MONITORS, AND ON PROCESSORS WITH NO HARDWARE
SWITCH REGISTER.

1.2 SYSTEM REQUIREMENTS.

HARDWARE REQUIREMENTS:

ANY Q-BUS PDP-11 FAMILY PROCESSOR
4K MEMORY - MINIMUM
A SPECIAL DATA WRAP AROUND CONNECTOR OR EQUIVALENT
(REQ'D IF DATA WRAP AROUND TESTS DESIRED)
IF CHANNEL 3 IS CONFIGURED AS THE CONSOLE, THE FOLLOWING
TESTS WILL BE BYPASSED ON CHANNEL 3:

TESTS 6 THRU 12
TESTS 14 THRU 17
TESTS 21 & 22

SOFTWARE REQUIREMENTS:

THIS DIAGNOSTIC IS DESIGNED TO RUN IN ANY OF THE FOLLOWING WAYS:
STAND ALONE
WITH APT MONITOR
WITH XXDP MONITOR (CHAINABLE IF RENAMED TO .BIC EXTENSION)

THIS DIAGNOSTIC IS NOT DESIGNED TO RUN WITH THE DIAGNOSTIC SUPERVISOR.

1.3 RELATED DOCUMENTS AND STANDARDS.

DIAGNOSTIC ENGINEERING STANDARDS AND CONVENTIONS	175-003-009-02
APT	MD-11-DZZMA
SYSMAC	MD-11-DZQAC

1.4 DIAGNOSTIC HIERARCHY PREREQUISITES.
NO SPECIAL DIAGNOSTICS ARE REQUIRED TO RUN BEFORE THIS, BUT THE PROCESSOR, MEMORY, AND BUS ARE ASSUMED TO BE FULLY OPERATIONAL.

1.5 ASSUMPTIONS.

THIS DIAGNOSTIC ASSUMES THAT THE OPERATOR HAS INITIALIZED LOCATION '$USWR' AND '$DEVM' TO THE PROPER VALUES. (SEE SEC. 2.4)

2.0 OPERATING INSTRUCTIONS.

2.1 LOADING AND STARTING PROCEDURES.

USE STANDARD PROCEDURE FOR PDP-11 ABSOLUTE BINARY FORMATTED MEDIA.

THERE ARE 2 STARTING LOCATIONS FOR THIS DIAGNOSTIC:

1. LOCATION 200 FOR ALL NORMAL TESTING STARTS & RESTARTS.

2. LOCATION 400 FOR INTERRUPT VECTOR DEBUG... OFFLINE ONLY.

THE 400 START SHOULD BE USED ONLY TO DEBUG FAULTY VECTOR ADDRESSES PUT OUT BY THE DV11-J DURING INTERRUPTS.
STARTING AT 400 WILL CAUSE THE ENTIRE VECTOR AREA, EXCEPT THAT NEEDED BY SYSMAC, TO BE OVERWRITTEN & TO POINT TO A COMMON SERVICE ROUTINE TO ENABLE LOOPING ON THE PROBLEM.

AS SOON AS TESTING STARTS, THE OPERATOR CAN CHANGE THE SWITCH REGISTER ONLY BY A 'BREAK' & MANUALLY LOADING LOCATION 176 (SWREG) WITH THE DESIRED CONTENTS (SEE SEC. 2.3) THEN DOING A 'P' TO PROCEED.

IT IS IMPORTANT THAT WHEN GOING BACK TO NORMAL TESTING A FRESH LOAD OF THE PROGRAM BE PERFORMED TO RESET THE VECTOR AREA.

THE USER CAN SELECT A SPECIFIC TEST TO BE EXECUTED BY SETTING BIT 8 IN SWREG AND THE TEST NUMBER (IN OCTAL) IN BITS <7:0>.
(NOTE: ALL TESTS PREVIOUS TO THE SELECTED ONE ARE EXECUTED WITHOUT ITERATIONS.)

2.2 SPECIAL ENVIRONMENTS.

THIS DIAGNOSTIC FOLLOWS THE STANDARD PROCEDURE FOR RUNNING UDER APT,XXDP MONITORS, AS DESCRIBED IN THEIR RESPECTIVE PROCEDURES MANUAL AND SYSMAC PACKAGE.

2.3 OPERATIONAL SWITCH SETTINGS

THE SOFTWARE SWITCH REGISTER (LOC. 176) IS USED FOR ALL OPERATIONAL SWITCH SETTINGS.
THIS CAN BE ACCOMPLISHED IN THE FOLLOWING WAY:

1) TYPE CONTROL G < G>; THIS WILL ALLOW THE TTY TO ENTER DATA INTO LOC. 176 AT SELECTED POINTS WITHIN THE PROGRAM.

2) THE MACHINE WILL THEN TYPE:' SWR=XXXXXX NEW=' (XXXXXX IS THE OCTAL CONTENTS OF THE SOFTWARE SWITCH REGISTER.)

3) AFTER THE 'NEW=' HAS BEEN TYPED THEN THE OPERATOR CAN DO ONE OF THE FOLLOWING AT THE TTY:

   A) TYPE A NUMBER TO BE LOADED INTO LOC. 176 FOLLOWED BY A <CR>. (ONLY NUMBERS BETWEEN 0-7 WILL BE ACCEPTED). LEADING ZEROS NEED NOT BE TYPED, AND IF MORE THAN 6 DIGITS ARE TYPED THE LAST 6 WILL BE USED. IF A <CR> IS THE FIRST KEY DEPRESSED THE SOFTWARE SWITCH REGISTER CONTENTS WILL NOT BE CHANGED.

   B) IF A CONTROL U < U> IS DEPRESSED THEN THE PROGRAM WILL SEND YOU BACK TO STEP 3.

   C) IF THE INPUT CHARACTER IS NOT ONE OF THE CHARACTERS MENTIONED ABOVE THEN A QUESTION MARK (?) WILL BE TYPED FOLLOWED BY A CARRAGE RETURN AND A LINE FEED SEQUENCE THEN PROCEED FROM STEP 3 (ERASING ALL PREVIOUS INPUT).

4) THE DIAGNOSTIC WILL CONTINUE ON TYPING <CR>.

SOFTWARE SWITCH REGISTER OPTIONS (SWREG)

```
BIT 15 SET = 100000 = HALT ON ERROR
    14 SET =  40000 = LOOP ON TEST  (TO BE USED ONLY WHILE TESTING IN PROGRESS)
    13 SET =  20000 = INHIBIT ERROR TYPEOUTS
    12 SET =  10000 = ENABLE PERFORMANCE REPORTS
    11 SET =   4000 = INHIBIT ITERATIONS
    10 SET =   2000 = BELL ON ERROR
     9 SET =   1000 = LOOP ON ERROR
     8 SET =    400 = LOOP ON TEST IN SWR<7:0>
     7:0   =        NUMBER OF TEST TO LOOP ON (USED WITH BIT 8)
                      (ALL TESTS PREVIOIUS TO THE SELECTED TEST
                      ARE EXECUTED FIRST WITH 1 ITERATION ONLY)
```

2.4 PROGRAM OPTIONS.

THIS PROGRAM REQUIRES THE ADDRESS OF THE FIRST RCSR (STORED AT '$BASE') AND ITS INTERRUPT VECTOR (STORED AT '$VECT1'). IT WILL BE ABLE TO ADDRESS ANY DLV11-J STARTING AT THE SPECIFIED BASE ADDRESS THRU 2 CONSECUTIVE MODULES.
EXAMPLES: $BASE: 176500 (DEFAULT)
$VECT1: 300 (DEFAULT)

LOCATION '$DEVM' IS USED AS A BIT MAP TO INDICATE WHICH UNIT NUMBERS ARE PRESENT AND WILL BE TESTED.

$DEVM (BITS 15 THRU 8 NOT USED)

								MODULE 2				MODULE 1				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	BIT
								3	2	1	0	3	2	1	0	CHANNEL

A BIT MAP CAN BE ENTERED AT '$DEVM' PRIOR TO STARTING THE PROGRAM. IF '$DEVM' IS LEFT AT ITS DEFAULT VALUE (0), THE PROGRAM WILL DO AUTO-SIZING BEGINNING AT THE DEVICE ADDRESS IN '$BASE' & WILL SETUP '$DEVM' ACCORDINGLY. IN EITHER CASE, THE PROGRAM WILL TYPE OUT A LIST OF ALL THE MODULES/CHANNELS TO BE TESTED BEFORE PROCEEDING.

NOTE: SIZING IS PERFORMED ONLY ONCE AT THE BEGINNING OF THE PROGRAM. IF $DEVM IS TO BE CHANGED, THE PROGRAM MUST BE RESTARTED AT 200 FOR IT TO BE EFFECTIVE. OTHERWISE, $DEVM WILL BE IGNORED.

OPTIONS

LOCATION $USWR CONTAINS ALL THE USER SELECTABLE OPTIONS. THE VALUES IN THIS WORD MUST CONFORM TO THE ACTUAL BOARD CONFIGURATION.

BIT POSITION	DEFINITION	$USWR DEFAULT VALUE	
0	# OF DATA BITS TRANSMITTED 0 = 7 BITS, 1 = 8 BITS	1	= 8 BITS
1	PARITY ENABLED	0	= NO
2	EVEN ODD PARITY	0	= ODD
3	BREAK GENERATION ENABLED	1	= YES
4	RUN DATA WRAP AROUND TESTS	1	= YES
<11:9>	CONSOLE DEVICE	1	= YES = CONSOLE ON MODULE 1, CHAN. 3

<11:9> = 1 = CONSOLE ON MODULE 1
<11:9> = 2 = CONSOLE ON MODULE 2

IMPORTANT NOTE: FOR DIAGNOSTIC PURPOSES, ALL CHANNELS MUST BE CONFIGURED THE SAME, EITHER 7 OR 8 BITS.

SUMMARY OF DEVICE ADDRESSES & VECTORS.

			DEV ADDRESS (RCSR)	VECTORS REC	VECTORS XMIT
MODULE #1	CHANNEL	0	$BASE+00	$VECT1+00	$VECT1+4
		1	+10	+10	+14
		2	+20	+20	+24
		3	+30	+30	+34
MODULE #2	CHANNEL	0	+40	+40	+44
		1	+50	+50	+54
		2	+60	+60	+64
		3	+70	+70	+74
CONSOLE DEVICE		3	$TKS	TKVEC=60	TPVEC=64

SUMMARY OF USER LOCATIONS & DEFAULTS.

	LOC	DEFAULT	
$BASE	1250	176500	
$VECT1	1244	300	
$TKS	1144	177560	FOR CONSOLE DEVICE ON MODULE UNDER TEST
TKVEC		60	FOR CONSOLE DEVICE ON MODULE UNDER TEST
TPVEC		64	FOR CONSOLE DEVICE ON MODULE UNDER TEST
$USWR	1220	1031	SEE SEC. 2.4
$DEVM	1252	0	SEE SEC. 2.4
SWREG	176	0	SEE SEC. 2.3

2.5 EXECUTION TIMES.

EXECUTION TIMES FOR AN LSI-11 PROCESSOR WITH ONE DLV11-J MODULE AT SHIPMENT CONFIGURATION:

CH. 0,1,2 AT 9600 BAUD.
CH. 3 (CONSOLE) AT 300 BAUD.

ARE: FIRST PASS- 30 SEC
ADDITIONAL PASSES 90 SEC

THE TEST TIME IS BAUD RATE DEPENDENT; HIGHER BAUD RATES RESULT IN SHORTER PASS TIMES.

2.6 POWER FAIL.

AUTO START FROM POWER FAIL IS NOT IMPLEMENTED IN THIS PROGRAM.

3.0 ERROR INFORMATION.

3.1 ERROR REPORTING PROCEDURE.

SINCE THIS DIAGNOSTIC WAS DESIGNED TO FIT IN 4K OF MEMORY THE ERROR TYPEOUT IS VERY BRIEF. THE FORMAT OF THE ERROR TYPEOUT IS AS FOLLOWS:

TEST#_____,ERROR#_____,PC=_____,ADDRESS=______,VECTOR=_____

WHERE ALL VALUES TYPED ARE OCTAL.
THE ADDRESS AND VECTOR REFER TO THE FAILING CHANNEL.
FOR FURTHER INFORMATION THE LISTING MUST BE CONSULTED.
BITS 15,13,10 AND 9 OF THE SWITCH REGISTER (SWREG) CONTROL THE SEQUENCE OF EVENTS AFTER AN ERROR IS CAUGHT.

BIT 15 SET: CAUSES THE PROGRAM TO HALT IN THE ERROR ROUTINE. IF THE PROGRAM IS CONTINUED, IT WILL PROCEED FROM WHERE IT HALTED.

BIT 13 SET: DISABLES THE PRINTING OF THE ERROR MESSAGE.

BIT 10 SET: CAUSES THE BELL TO RING ON ERROR.

BIT 9 SET: CAUSES THE DIAGNOSTIC TO LOOP FROM BEGINNING OF TEST TO ERROR.

THE ERROR ROUTINE SUPPORTS THE CONTROL G < G> FUNCTION.
REFER TO SECTION 2.3 FOR DETAILS.

3.2 ERROR HALTS.

THE ONLY HALT IN THIS DIAGNOSTIC IS IN THE ERROR ROUTINE, AND IS EXECUTED ONLY IF BIT 15 OF THE SWITCH REGISTER (SWREG) IS SET WHEN AN ERROR OCCURS.

4.0 PERFORMANCE AND PROGRESS REPORTS.

4.1 PERFORMANCE REPORTS. (BIT 12 SET IN THE SWITCH REGISTER 'SWREG')

AS EACH CHANNEL (4 CHANNELS/DLV11-J) COMPLETES ONE PASS OF THE DIAGNOSTIC THE FOLLOWING ITEMS ARE TYPED:

'CSR:_____' : THE BASE ADDRESS OF THE LINE UNDER TEST
'VECTOR:__' : THE ASSOCIATED VECTOR
'ERRORS:__' : THE TOTAL NUMBER OF ERRORS ON THIS DEVICE ON THIS PASS.

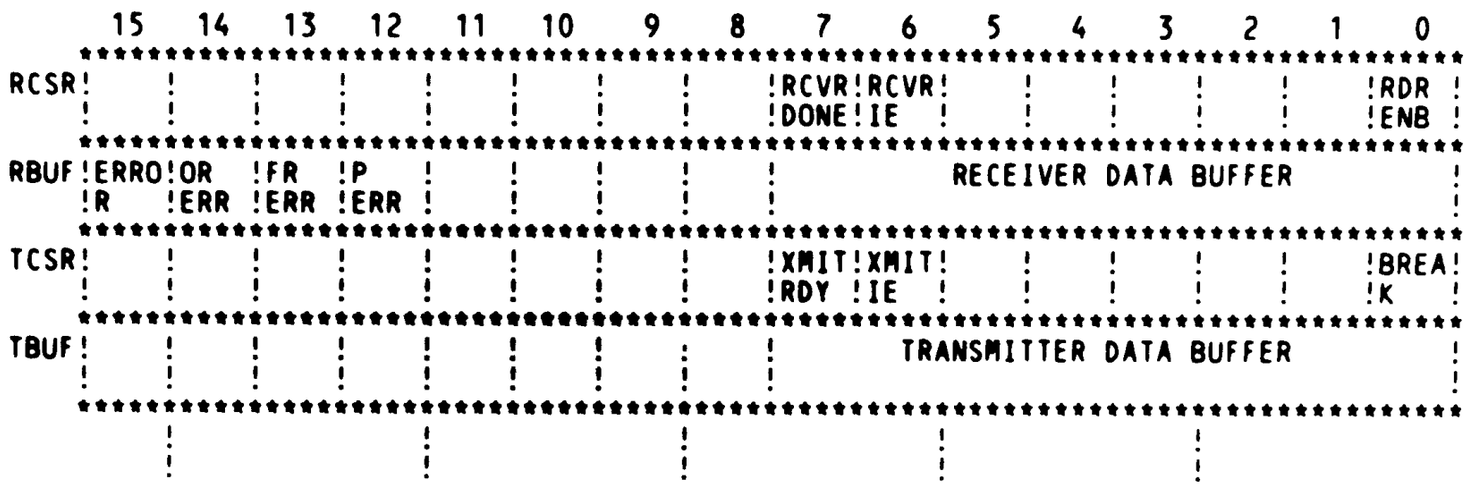
AFTER ALL MODULES & CHANNELS TO BE TESTED HAVE BEEN EXERCISED, AN END PASS STATEMENT IS TYPED:

"END PASS#_____."

EXAMPLE OF PRINTOUT ASSUMING: 2 DLV11-J MODULES
ALL CHANNELS ENABLED
SEPERATE CONSOLE DEVICE
NO ERRORS

```
CVDLAA0 DLV11-J TEST

SWR= 000000     NEW= 10000<CR>                  (ENABLE PERFORMANCE REPORTS)

WILL TEST:

MODULE 1  CHANNEL 0  1  2  3
MODULE 2  CHANNEL 0  1  2  3

        ** PHASE 1 SUMMARY **
CSR: 176500, VECTOR: 000300, ERRORS: 0
CSR: 176510, VECTOR: 000310, ERRORS: 0
CSR: 176520, VECTOR: 000320, ERRORS: 0
CSR: 176530, VECTOR: 000330, ERRORS: 0

        ** PHASE 2 SUMMARY **
CSR: 176500, VECTOR: 000300, ERRORS: 0

        ** PHASE 1 SUMMARY **
CSR: 176540, VECTOR: 000340, ERRORS: 0
CSR: 176550, VECTOR: 000350, ERRORS: 0
CSR: 176560, VECTOR: 000360, ERRORS: 0
CSR: 176570, VECTOR: 000370, ERRORS: 0

        ** PHASE 2 SUMMARY **
CSR: 176540, VECTOR: 000340, ERRORS: 0

END PASS #      1
```

5.0 DEVICE REGISTERS.



NOTES:

1. RCSR AT BASE ADDRESS ($BASE)
   RBUF AT $BASE+2
   TCSR AT $BASE+4
   TBUF AT $BASE+6

2. BLANK BOXES INDICATE UNUSED AND RESERVED BIT POSITIONS. SEE THE LISTING FOR AN EXPLANATION OF THE BITS.

3. ORERR = OVERRUN ERROR
   FRERR = FRAMING ERROR
   PERR = PARITY ERROR

6.0 SUMMARY OF TESTS AND SPECIAL SUBROUTINES.

PHASE 1 TESTS

TEST 1 ADDRESSABILITY

THIS TEST VERIFIES THAT ALL 4 REGISTERS OF THE CHANNEL UNDER TEST RESPOND TO THEIR ADDRESSES.

THE FOLLOWING 3 TESTS TEST ALL 'READ WRITE' BITS

TEST 2 BREAK - TCSR 0 SET, CLEAR, RESET

TEST 3 XMITIE - TCSR 6 SET, CLEAR, RESET

TEST 4 RCVRIE - RCSR 6 SET, CLEAR, RESET

TEST 5 XMITRDY - TCSR 7 - IS SET BY INIT

TEST 6 XMIT RDY - TCSR 7 - CLEARS WHEN TBUF IS LOADED WITH A CHARACTER AND THAT IT SETS WITHIN A REASONABLE AMOUNT OF TIME.

TEST 7 OUTPUTTING A CHAR FROM TBUF (WITH WRAP AROUND CONNECTED) RESULTS IN RCVRDONE SETTING WITHIN A REASONABLE AMOUNT OF TIME AND THAT RESET CLEARS THE BIT.

TEST 10 RCVRDONE IS CLEARED BY SETTING READER ENABLE

TEST 11 RCVRDONE IS CLEARED BY READING RBUF

TEST 12 OVERRUN & ERROR BIT - RBUF 14

TEST 13 TRANSMITTER INTERRUPT LOGIC TEST
---- --

LOGICALLLY THIS IS 4 SEPARATE TESTS
A) DOES TRANSMITTER INTERRUPT LOGIC WORK
B) AT PRIORITY OF 0
C) AND ONLY ONCE
D) BUT NOT WITH INTERRUPT ENABLE CLEAR

TEST 14 RECEIVER INTERRUPT LOGIC TEST THIS TEST COVERS ALL
---- -- OF THE RECEIVER SIDE OF THE INTERRUPT LOGIC IN
CHARACTER MODE.

TEST 15 TEST DATA WRAP AROUND: FLAG MODE.
---- --

TEST 16 TEST DATA WRAP AROUND: INTERRUPT MODE.
---- --

TEST 17 TEST BREAK GENERATION LOGIC TRANSMIT KNOWN CHAR
---- --

A) TRANSMIT KNOWN CHAR WITH BREAK SET
AND COMPARE RECEIVED WITH 0

B) TEST FOR FRAMING ERROR ON BREAK

C) IF PARITY IS ENABLED AND ODD PARITY IS SELECTED,
CHECK TO BE SURE PARITY ERROR WAS GENERATED

D) IF PARITY IS ENABLED AND EVEN PARITY IS SELECTED,
CHECK TO BE SURE NO PARITY ERROR OCCURRED

TEST 20 NOT A TEST - SEND BACK TO LOOP

PHASE 2 TESTS

TEST 21 TEST THAT CHANNELS INTERRUPT AT ASSIGNED PRIORITY
---- --

TEST 22 TEST DATA TRANSFERS WITH ALL ACTIVE LINES INTERRUPTING.
---- --

%

```
                        .TITLE  CVDLAA0 DLV11-J TEST
                        ;*COPYRIGHT (C) 1978
                        ;*DIGITAL EQUIPMENT CORP.
                        ;*MAYNARD, MASS. 01754
                        ;*
                        ;*PROGRAM BY GREGORY GLEZMAN & GARY PAPAZIAN
                        ;*
                        ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                        ;*PACKAGE (MAINDEC-11-DZQAC-C3), JAN 19, 1977.
                        ;*

                        .SBTTL  OPERATIONAL SWITCH SETTINGS
                        ;*
                        ;*      SWITCH                  USE
                        ;*      ------          --------------------
                        ;*        15            HALT ON ERROR
                        ;*        14            LOOP ON TEST
                        ;*        13            INHIBIT ERROR TYPEOUTS
                        ;*        11            INHIBIT ITERATIONS
                        ;*        10            BELL ON ERROR
                        ;*         9            LOOP ON ERROR
                        ;*         8            LOOP ON TEST IN SWR<7:0>
                        .SBTTL  BASIC DEFINITIONS

                        ;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
001100                  STACK=  1100
                        .EQUIV  EMT,ERROR       ;;BASIC DEFINITION OF ERROR CALL
                        .EQUIV  IOT,SCOPE       ;;BASIC DEFINITION OF SCOPE CALL

                        ;*MISCELLANEOUS DEFINITIONS
000011                  HT=     11              ;;CODE FOR HORIZONTAL TAB
000012                  LF=     12              ;;CODE FOR LINE FEED
000015                  CR=     15              ;;CODE FOR CARRIAGE RETURN
000200                  CRLF=   200             ;;CODE FOR CARRIAGE RETURN-LINE FEED
177776                  PS=     177776          ;;PROCESSOR STATUS WORD
                        .EQUIV  PS,PSW
177774                  STKLMT= 177774          ;;STACK LIMIT REGISTER
177772                  PIRQ=   177772          ;;PROGRAM INTERRUPT REQUEST REGISTER
177570                  DSWR=   177570          ;;HARDWARE SWITCH REGISTER
177570                  DDISP=  177570          ;;HARDWARE DISPLAY REGISTER

                        ;*GENERAL PURPOSE REGISTER DEFINITIONS
000000                  R0=     %0              ;;GENERAL REGISTER
000001                  R1=     %1              ;;GENERAL REGISTER
000002                  R2=     %2              ;;GENERAL REGISTER
000003                  R3=     %3              ;;GENERAL REGISTER
000004                  R4=     %4              ;;GENERAL REGISTER
000005                  R5=     %5              ;;GENERAL REGISTER
000006                  R6=     %6              ;;GENERAL REGISTER
000007                  R7=     %7              ;;GENERAL REGISTER
000006                  SP=     %6              ;;STACK POINTER
000007                  PC=     %7              ;;PROGRAM COUNTER

                        ;*PRIORITY LEVEL DEFINITIONS
000000                  PR0=    0               ;;PRIORITY LEVEL 0
```

```
                    
000040              PR1=    40              ;;PRIORITY LEVEL 1
000100              PR2=    100             ;;PRIORITY LEVEL 2
000140              PR3=    140             ;;PRIORITY LEVEL 3
000200              PR4=    200             ;;PRIORITY LEVEL 4
000240              PR5=    240             ;;PRIORITY LEVEL 5
000300              PR6=    300             ;;PRIORITY LEVEL 6
000340              PR7=    340             ;;PRIORITY LEVEL 7

                    ;*"SWITCH REGISTER" SWITCH DEFINITIONS
100000              SW15=   100000
040000              SW14=   40000
020000              SW13=   20000
010000              SW12=   10000
004000              SW11=   4000
002000              SW10=   2000
001000              SW09=   1000
000400              SW08=   400
000200              SW07=   200
000100              SW06=   100
000040              SW05=   40
000020              SW04=   20
000010              SW03=   10
000004              SW02=   4
000002              SW01=   2
000001              SW00=   1
                    .EQUIV  SW09,SW9
                    .EQUIV  SW08,SW8
                    .EQUIV  SW07,SW7
                    .EQUIV  SW06,SW6
                    .EQUIV  SW05,SW5
                    .EQUIV  SW04,SW4
                    .EQUIV  SW03,SW3
                    .EQUIV  SW02,SW2
                    .EQUIV  SW01,SW1
                    .EQUIV  SW00,SW0

                    ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
100000              BIT15=  100000
040000              BIT14=  40000
020000              BIT13=  20000
010000              BIT12=  10000
004000              BIT11=  4000
002000              BIT10=  2000
001000              BIT09=  1000
000400              BIT08=  400
000200              BIT07=  200
000100              BIT06=  100
000040              BIT05=  40
000020              BIT04=  20
000010              BIT03=  10
000004              BIT02=  4
000002              BIT01=  2
000001              BIT00=  1
                    .EQUIV  BIT09,BIT9
                    .EQUIV  BIT08,BIT8
                    .EQUIV  BIT07,BIT7
```

```
                        .EQUIV  BIT06,BIT6
                        .EQUIV  BIT05,BIT5
                        .EQUIV  BIT04,BIT4
                        .EQUIV  BIT03,BIT3
                        .EQUIV  BIT02,BIT2
                        .EQUIV  BIT01,BIT1
                        .EQUIV  BIT00,BIT0

                        ;*BASIC "CPU" TRAP VECTOR ADDRESSES
000004                  ERRVEC= 4               ;;TIME OUT AND OTHER ERRORS
000010                  RESVEC= 10              ;;RESERVED AND ILLEGAL INSTRUCTIONS
000014                  TBITVEC=14              ;;"T" BIT
000014                  TRTVEC= 14              ;;TRACE TRAP
000014                  BPTVEC= 14              ;;BREAKPOINT TRAP (BPT)
000020                  IOTVEC= 20              ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
000024                  PWRVEC= 24              ;;POWER FAIL
000030                  EMTVEC= 30              ;;EMULATOR TRAP (EMT) **ERROR**
000034                  TRAPVEC=34              ;;"TRAP" TRAP
000060                  TKVEC=  60              ;;TTY KEYBOARD VECTOR
000064                  TPVEC=  64              ;;TTY PRINTER VECTOR
000240                  PIRQVEC=240             ;;PROGRAM INTERRUPT REQUEST VECTOR


000004                  ILLMEM= 4
000001                  ADRS=   R1
176500                  DLADDR= 176500

                        ; THE FOLLOWING DEFINITIONS APPLY TO THE GLOBAL SUBS
177777                  SET= -1
000000                  CLR=  0
000001                  TRUE= 1
```

```
                        ;;*****************************************************************
                        ; RCSR REGISTER BIT NAMES
                        ;;*****************************************************************
                        ; UNUSED       BIT15
                        ; UNUSED       BIT14
                        ; UNUSED       BIT13
                        ; UNUSED       BIT12
                        ; UNUSED       BIT11
                        ; UNUSED       BIT10
                        ; UNUSED       BIT09
                        ; UNUSED       BIT08
000200                  RCVRDONE=      BIT07        ; RECEIVER DONE
000100                  RCVRIE=        BIT06        ; RECEIVER INTERRUPT ENABLE
                        ; UNUSED       BIT05
                        ; UNUSED       BIT04
                        ; UNUSED       BIT03
                        ; UNUSED       BIT02
                        ; UNUSED       BIT01
000001                  RDRRUN=        BIT00        ; READER RUN

                        ;;*****************************************************************
                        ; RBUF REGISTER BIT NAMES
                        ;;*****************************************************************
100000                  ERROR=         BIT15        ; ERROR INDICATOR
040000                  ORERR=         BIT14        ; OVERRUN ERROR
020000                  FRERR=         BIT13        ; FRAMING ERROR
010000                  PERR=          BIT12        ; PARITY ERROR
                        ; UNUSED       BIT11
                        ; UNUSED       BIT10
                        ; UNUSED       BIT09
                        ; UNUSED       BIT08
000200                  RDATA7=        BIT07        ;
000100                  RDATA6=        BIT06        ;  !
000040                  RDATA5=        BIT05        ;  !
000020                  RDATA4=        BIT04        ;      RECEIVED DATA BITS
000010                  RDATA3=        BIT03        ;   /
000004                  RDATA2=        BIT02        ;  !
000002                  RDATA1=        BIT01        ;  !
000001                  RDATA0=        BIT00        ; /
```

```
                        ;;*****************************************************************
                        ;               TCSR REGISTER BIT NAMES
                        ;;*****************************************************************
                        ;       UNUSED          BIT15
                        ;       UNUSED          BIT14
                        ;       UNUSED          BIT13
                        ;       UNUSED          BIT12
                        ;       UNUSED          BIT11
                        ;       UNUSED          BIT10
                        ;       UNUSED          BIT09
                        ;       UNUSED          BIT08
000200                  XMITRDY=        BIT07   ; TRANSMITTER READY
000100                  XMITIE=         BIT06   ; TRANSMITTER INTERRUPT ENABLE
                        ;       UNUSED          BIT05
                        ;       UNUSED          BIT04
                        ;       UNUSED          BIT03
                        ;       UNUSED          BIT02
                        ;       UNUSED          BIT01
000001                  BREAK=          BIT00   ; SEND BREAK (CONTINUOUS SPACE)

                        ;;*****************************************************************
                        ;               TBUF REGISTER BIT NAMES
                        ;;*****************************************************************
                        ;       UNUSED          BIT15
                        ;       UNUSED          BIT14
                        ;       UNUSED          BIT13
                        ;       UNUSED          BIT12
                        ;       UNUSED          BIT11
                        ;       UNUSED          BIT10
                        ;       UNUSED          BIT09
                        ;       UNUSED          BIT08
000200                  TDATA7=         BIT07   ;
000100                  TDATA6=         BIT06   ; !
000040                  TDATA5=         BIT05   ; .
000020                  TDATA4=         BIT04   ;    TRANSMITTER DATA BUFFER
000010                  TDATA3=         BIT03   ;  /
000004                  TDATA2=         BIT02   ; !
000002                  TDATA1=         BIT01   ; !
000001                  TDATA0=         BIT00   ; /

                        ;;*****************************************************************
                        ;       FLAG BITS TO BE USE OR CLEARED IN $USWR.

000002                      PARITY   =  2
000004                      EVENODD  =  4
000010                      BRK      =  10
000020                      WRAP     =  20
```

```
                        ;;*****************************************************************
                        .SBTTL  TRAP CATCHER

                                .=0
                        ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
                        ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
                        ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
                                .=174
000174  000000          DISPREG: .WORD  0               ;;SOFTWARE DISPLAY REGISTER
000176  000000          SWREG:   .WORD  0               ;;SOFTWARE SWITCH REGISTER
                        .SBTTL  STARTING ADDRESS(ES)
000200  000137  001334          JMP     @#START ;;JUMP TO STARTING ADDRESS OF PROGRAM



                        ;
                        ; OFFLINE TESTING FOR INTERRUPT VECTOR PROBLEMS
                        ;
                                .=400

000400  005000          1$:     CLR     R0
000402  012720  013742          MOV     #INTSRV,(R0)+   ;SET INTR HANDLER PTR
000406  012720  000340          MOV     #PR7,(R0)+      ;SET PRIORITY
000412  020027  000400          CMP     R0,#400         ;ALL DONE?
000416  001371                  BNE     1$              ;BR IF NO
000420  005037  000176          CLR     176             ;CLEAN UP SWREG
000424  000137  001334          JMP     START           ;GO DO IT
```

```
                                .SBTTL  ACT11 HOOKS

                                ;;*****************************************************************
                                ;HOOKS REQUIRED BY ACT11
        000430                          $SVPC=.                 ;SAVE PC
        000046                          .=46
000046  014602                          $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
        000052                          .=52
000052  000000                          .WORD   0               ;;2)SET LOC.52 TO ZERO
        000430                          .=$SVPC                 ;; RESTORE PC
        001000                  .-1000
                                .SBTTL  APT PARAMETER BLOCK

                                ;;*****************************************************************
                                ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
                                ;;*****************************************************************
        001000                          .$X=.   ;;SAVE CURRENT LOCATION
        000024                          .=24    ;;SET POWER FAIL TO POINT TO START OF PROGRAM
000024  000200                          200     ;;FOR APT START UP
        000044                          .=44    ;;POINT TO APT INDIRECT ADDRESS PNTR.
000044  001000                          $APTHDR ;;POINT TO APT HEADER BLOCK
        001000                          .=.$X   ;;RESET LOCATION COUNTER
                                ;;*****************************************************************
                                ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
                                ;INTERFACE SPEC.

001000                          $APTHD:
001000  000000                  $HIBTS: .WORD   0       ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
001002  001174                  $MBADR: .WORD   $MAIL   ;;ADDRESS OF APT MAILBOX (BITS 0-15)
001004  000031                  $TSTM:  .WORD   25.     ;;RUN TIM OF LONGEST TEST
001006  000144                  $PASTM: .WORD   100.    ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
001010  000144                  $UNITM: .WORD   100.    ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
001012  000030                          .WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
```

```
                        .SBTTL  COMMON TAGS

                        ;;*****************************************************************
                        ;*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
                        ;*USED IN THE PROGRAM.

            001100              .=1100
001100                  $CMTAG:                         ;;START OF COMMON TAGS
001100  000000                  .WORD   0
001102     000          $TSTNM: .BYTE   0               ;;CONTAINS THE TEST NUMBER
001103     000          $ERFLG: .BYTE   0               ;;CONTAINS ERROR FLAG
001104  000000          $ICNT:  .WORD   0               ;;CONTAINS SUBTEST ITERATION COUNT
001106  000000          $LPADR: .WORD   0               ;;CONTAINS SCOPE LOOP ADDRESS
001110  000000          $LPERR: .WORD   0               ;;CONTAINS SCOPE RETURN FOR ERRORS
001112  000000          $ERTTL: .WORD   0               ;;CONTAINS TOTAL ERRORS DETECTED
001114     000          $ITEMB: .BYTE   0               ;;CONTAINS ITEM CONTROL BYTE
001115     001          $ERMAX: .BYTE   1               ;;CONTAINS MAX. ERRORS PER TEST
001116  000000          $ERRPC: .WORD   0               ;;CONTAINS PC OF LAST ERROR INSTRUCTION
001120  000000          $GDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'GOOD' DATA
001122  000000          $BDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'BAD' DATA
001124  000000          $GDDAT: .WORD   0               ;;CONTAINS 'GOOD' DATA
001126  000000          $BDDAT: .WORD   0               ;;CONTAINS 'BAD' DATA
001130  000000                  .WORD   0               ;;RESERVED--NOT TO BE USED
001132  000000                  .WORD   0
001134     000          $AUTOB: .BYTE   0               ;;AUTOMATIC MODE INDICATOR
001135     000          $INTAG: .BYTE   0               ;;INTERRUPT MODE INDICATOR
001136  000000                  .WORD   0
001140  177570          SWR:    .WORD   DSWR            ;;ADDRESS OF SWITCH REGISTER
001142  177570          DISPLAY: .WORD  DDISP           ;;ADDRESS OF DISPLAY REGISTER
001144  177560          $TKS:   177560                  ;;TTY KBD STATUS
001146  177562          $TKB:   177562                  ;;TTY KBD BUFFER
001150  177564          $TPS:   177564                  ;;TTY PRINTER STATUS REG. ADDRESS
001152  177566          $TPB:   177566                  ;;TTY PRINTER BUFFER REG. ADDRESS
001154     000          $NULL:  .BYTE   0               ;;CONTAINS NULL CHARACTER FOR FILLS
001155     002          $FILLS: .BYTE   2               ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
001156     012          $FILLC: .BYTE   12              ;;INSERT FILL CHARS. AFTER A "LINE FEED"
001157     000          $TPFLG: .BYTE   0               ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
001160  000000          $TIMES: 0                       ;;MAX. NUMBER OF ITERATIONS
001162  000000          $ESCAPE:0                       ;;ESCAPE ON ERROR ADDRESS
001164  177607  000377  $BELL:  .ASCIZ  <207><377><377> ;;CODE FOR BELL
001170     077          $QUES:  .ASCII  /?/             ;;QUESTION MARK
001171     015          $CRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
001172  000012          $LF:    .ASCIZ  <12>            ;;LINE FEED
                        ;;*****************************************************************
                        .SBTTL  APT MAILBOX-ETABLE

                        ;;*****************************************************************
                        .EVEN
001174                  $MAIL:                          ;;APT MAILBOX
001174  000000          $MSGTY: .WORD   AMSGTY          ;;MESSAGE TYPE CODE
001176  000000          $FATAL: .WORD   AFATAL          ;;FATAL ERROR NUMBER
001200  000000          $TESTN: .WORD   ATESTN          ;;TEST NUMBER
001202  000000          $PASS:  .WORD   APASS           ;;PASS COUNT
001204  000000          $DEVCT: .WORD   ADEVCT          ;;DEVICE COUNT
001206  000000          $UNIT:  .WORD   AUNIT           ;;I/O UNIT NUMBER
001210  000000          $MSGAD: .WORD   AMSGAD          ;;MESSAGE ADDRESS
```

```
941  001212  000000                $MSGLG: .WORD   AMSGLG  ;;MESSAGE LENGTH
942  001214                        $ETABLE:                ;;APT ENVIRONMENT TABLE
943  001214     000                $ENV:   .BYTE   AENV    ;;ENVIRONMENT BYTE
944  001215     000                $ENVM:  .BYTE   AENVM   ;;ENVIRONMENT MODE BITS
945  001216  000000                $SWREG: .WORD   ASWREG  ;;APT SWITCH REGISTER
946  001220  001031                $USWR:  .WORD   AUSWR   ;;USER SWITCHES
947  001222  000000                $CPUOP: .WORD   ACPUOP  ;;CPU TYPE,OPTIONS
948                                ;*                      BITS 15-11=CPU TYPE
949                                ;*                              11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
950                                ;*                              11/70=06,PDQ=07,Q=10
951                                ;*                      BIT 10=REAL TIME CLOCK
952                                ;*                      BIT  9=FLOATING POINT PROCESSOR
953                                ;*                      BIT  8=MEMORY MANAGEMENT
954  001224     000                $MAMS1: .BYTE   AMAMS1  ;;HIGH ADDRESS,M.S. BYTE
955  001225     000                $MTYP1: .BYTE   AMTYP1  ;;MEM. TYPE,BLK#1
956                                ;*                      MEM.TYPE BYTE   --  (HIGH BYTE)
957                                ;*                              900 NSEC CORE=001
958                                ;*                              300 NSEC BIPOLAR=002
959                                ;*                              500 NSEC MOS=003
960  001226  000000                $MADR1: .WORD   AMADR1  ;;HIGH ADDRESS,BLK#1
961                                ;*                        MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE
962  001230     000                $MAMS2: .BYTE   AMAMS2  ;;HIGH ADDRESS,M.S. BYTE
963  001231     000                $MTYP2: .BYTE   AMTYP2  ;;MEM.TYPE,BLK#2
964  001232  000000                $MADR2: .WORD   AMADR2  ;;MEM.LAST ADDRESS,BLK#2
965  001234     000                $MAMS3: .BYTE   AMAMS3  ;;HIGH ADDRESS,M.S.BYTE
966  001235     000                $MTYP3: .BYTE   AMTYP3  ;;MEM.TYPE,BLK#3
967  001236  000000                $MADR3: .WORD   AMADR3  ;;MEM.LAST ADDRESS,BLK#3
968  001240     000                $MAMS4: .BYTE   AMAMS4  ;;HIGH ADDRESS,M.S.BYTE
969  001241     000                $MTYP4: .BYTE   AMTYP4  ;;MEM.TYPE,BLK#4
970  001242  000000                $MADR4: .WORD   AMADR4  ;;MEM.LAST ADDRESS,BLK#4
971  001244  000300                $VECT1: .WORD   AVECT1  ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
972  001246  000000                $VECT2: .WORD   AVECT2  ;;INTERRUPT VECTOR#2BUS PRIORITY#2
973  001250  176500                $BASE:  .WORD   ABASE   ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
974  001252  000000                $DEVM:  .WORD   ADEVM   ;;DEVICE MAP
975  001254                        $ETEND:
976                                .MEXIT
```

```
                                .SBTTL  ERROR POINTER TABLE

                                ;*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
                                ;*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
                                ;*LOCATION $ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
                                ;*NOTE1:        IF $ITEMB IS 0 THE ONLY PERTINENT DATA IS ($ERRPC).
                                ;*NOTE2:        EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:

                                ;*      EM              ;;POINTS TO THE ERROR MESSAGE
                                ;*      DH              ;;POINTS TO THE DATA HEADER
                                ;*      DT              ;;POINTS TO THE DATA
                                ;*      DF              ;;POINTS TO THE DATA FORMAT


001254                          $ERRTB:
                                ;;          GLOBAL DATA
001254  176500                      DLADD:  DLADDR
001256  000300                      DLVEC:  300
001260  176500                      RCSR:   DLADDR  +  0
001262  176502                      RBUF:   DLADDR  +  2
001264  176504                      TCSR:   DLADDR  +  4
001266  176506                      TBUF:   DLADDR  +  6
001270  000000                      I:      0
001272  000020                      .BLKW 20    ;FOR R5 STACK
001332  000000                      R5STACK:    .WORD 0
```

```
1002  001334                          START:
1003                                  .SBTTL  INITIALIZE THE COMMON TAGS
1004                                  ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
1005  001334  012706  001100                  MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
1006  001340  005026                          CLR     (R6)+           ;;CLEAR MEMORY LOCATION
1007  001342  022706  001140                  CMP     #SWR,R6 ;;DONE?
1008  001346  001374                          BNE     .-6             ;;LOOP BACK IF NO
1009  001350  012706  001100                  MOV     #STACK,SP       ;;SETUP THE STACK POINTER
1010                                  ;;INITIALIZE A FEW VECTORS
1011  001354  012737  016342  000020          MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
1012  001362  012737  000340  000022          MOV     #340,@#IOTVEC+2 ;;LEVEL 7
1013  001370  012737  016142  000030          MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
1014  001376  012737  000340  000032          MOV     #340,@#EMTVEC+2 ;;LEVEL 7
1015  001404  012737  017274  000034          MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
1016  001412  012737  000340  000036          MOV     #340,@#TRAPVEC+2;LEVEL 7
1017  001420  013737  014550  014542          MOV     $ENDCT,$EOPCT   ;;SETUP END-OF-PROGRAM COUNTER
1018  001426  005037  001160                  CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
1019  001432  005037  001162                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
1020  001436  112737  000001  001115          MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
1021  001444  012737  001444  001106          MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
1022  001452  012737  001452  001110          MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
1023                                  ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
1024                                  ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
1025  001460  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
1026  001464  012737  001520  000004          MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
1027  001472  012737  177570  001140          MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
1028  001500  012737  177570  001142          MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
1029  001506  022777  177777  177424          CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
1030  001514  001012                          BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
1031                                                                  ;;AND  THE HARDWARE SWR IS NOT = -1
1032  001516  000403                          BR      65$             ;;BRANCH IF NO TIMEOUT
1033  001520  012716  001526          64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
1034  001524  000002                          RTI
1035  001526  012737  000176  001140  65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
1036  001534  012737  000174  001142          MOV     #DISPREG,DISPLAY
1037  001542  012637  000004          66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR
1038
1039  001546  005037  001202                  CLR     $PASS           ;;CLEAR PASS COUNT
1040  001552  132737  000200  001215          BITB    #APTSIZE,$ENVM  ;;TEST USER SIZE UNDER APT
1041  001560  001403                          BEQ     67$             ;;YES,USE NON-APT SWITCH
1042  001562  012737  001216  001140          MOV     #$SWREG,SWR     ;;NO,USE APT SWITCH REGISTER
1043  001570                          67$:
1044                                  .SBTTL  TYPE PROGRAM NAME
1045                                  ;;TYPE THE NAME OF THE PROGRAM IF FIRST PASS
1046  001570  005227  177777                  INC     #-1             ;;FIRST TIME?
1047  001574  001043                          BNE     68$             ;;BRANCH IF NO
1048  001576  022737  014602  000042          CMP     #$ENDAD,@#42    ;;ACT-11?
1049  001604  001437                          BEQ     68$             ;;BRANCH IF YES
1050  001606  104401  001654                  TYPE    ,69$            ;;TYPE ASCIZ STRING
1051                                  .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
1052  001612  005737  000042                  TST     @#42            ;;ARE WE RUNNING UNDER XXDP/ACT?
1053  001616  001012                          BNE     70$             ;;BRANCH IF YES
1054  001620  123727  001214  000001          CMPB    $ENV,#1         ;;ARE WE RUNNING UNDER APT?
1055  001626  001406                          BEQ     70$             ;;BRANCH IF YES
1056  001630  023727  001140  000176          CMP     SWR,#SWREG      ;;SOFTWARE SWITCH REG SELECTED?
1057  001636  001005                          BNE     71$             ;;BRANCH IF NO
```

```
 1058  001640  104406                          GTSWR                   ;;GET SOFT-SWR SETTINGS
 1059  001642  000403                          BR      71$
 1060  001644  112737  000001  001134  70$:    MOVB    #1,$AUTOB       ;;SET AUTO-MODE INDICATOR
 1061  001652                          71$:
 1062  001652  000414                          BR      68$             ;;GET OVER THE ASCIZ
 1063                                  ;;69$:   .ASCIZ  <CRLF>*CVDLAA0 DLV11-J TEST*<CRLF>
 1064  001704                          68$:
```

```
1065
1066
1067  001704  004737  013012                  JSR     PC,SIZE         ;SIZE THE BUS IF $DEVM = 0
1068
1069  001710  005037  014320                  CLR     PHASE2          ;INIT FOR TESTS
1070  001714  012737  000001  001206          MOV     #1,$UNIT
1071  001722  005037  014316                  CLR     CHAN
1072  001726  005037  014322                  CLR     P1CNT
1073
1074  001732  012706  001100          LOOP:   MOV     #STACK,SP       ;RESET STACK POINTER
1075  001736  004737  013752                  JSR     PC,CYCLE        ;SETUP MODULE AND CHANNEL
1076                                                                  ;       ADRS+2 -> NEXT VECTOR
1077                                                                          ;GET UNIT ADDRESS
1078  001742                                                                  LET     DLADD := (ADRS)+
1079  001742  012137  001254                  MOV     (ADRS)+,DLADD
1080
1081  001746                                                                  ;GET UNIT VECTOR
1082  001746  011137  001256                  MOV     (ADRS),DLVEC            LET     DLVEC := (ADRS)
1083
1084  001752                                                                  ;RCSR = DLADD  +  0
1085  001752  013737  001254  001260          MOV     DLADD,RCSR              LET     RCSR  := DLADD
1086  001760
1087  001760  013737  001254  001262          MOV     DLADD,RBUF              LET     RBUF  := DLADD + #2
1088  001766  062737  000002  001262          ADD     #2,RBUF
1089  001774
1090  001774  013737  001254  001264          MOV     DLADD,TCSR              LET     TCSR  := DLADD + #4
1091  002002  062737  000004  001264          ADD     #4,TCSR
1092  002010
1093  002010  013737  001254  001266          MOV     DLADD,TBUF              LET     TBUF  := DLADD + #6
1094  002016  062737  000006  001266          ADD     #6,TBUF
1095  002024
1096  002024  012705  001332                  MOV     #R5STACK,R5             LET     R5    := #R5STACK
1097
1098  002030  012700  001000                  MOV     #1000,R0        ;DELAY TO ALLOW PRINT TO FINISH
1099  002034  077001                          SOB     R0,.            ;BEFORE RESET
1100  002036                                                          BRESET
1101  002036  000005                          RESET
1102  002040                                                                  LET     $DEVCT := $DEVCT + #1
1103  002040  005237  001204                  INC     $DEVCT
1104  002044                                                                  IFB     PHASE2  EQ  #TRUE  THEN
1105  002044  123727  014320  000001          CMPB    PHASE2,#TRUE
1106  002052  001003                          BNE     $1
1107  002054  000137  007112                  JMP     MODTST                      ;GO TO PHASE 2 TESTING
1108  002060                                                                  ELSE
1109  002060  000402                          BR      $2
1110  002062                           $1:
1111  002062                                                                     LET $TSTNM := #0
1112  002062  005037  001102                  CLR     $TSTNM
1113  002066                                                                  ENDIF
1114  002066                           $2:
```

```
1115                                    ;;*****************************************************************
1116                                    ;*TEST 1        ADDRESSABILITY
1117                                    ;*
1118                                    ;*      THIS TEST VERIFIES THAT ALL 4 REGISTERS OF THE CHANNEL UNDER TEST
1119                                    ;*      RESPOND TO THEIR ADDRESSES.
1120                                    ;*
1121                                    ;;*****************************************************************
1122  002066  000004                   TST1:   SCOPE
1123  002070  012737  000002  001160           MOV     #2,$TIMES       ;;DO 2 ITERATIONS
1124  002076  012737  000001  001200           MOV     #1,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
1125  002104                                                                   LET     ADRS := DLADD
1126  002104  013701  001254                   MOV     DLADD,ADRS
1127                                                                             ; SET UP INTERRUPT
1128  002110                                                                   SETVEC  #ILLMEM,#INTSRV,#PR7
1129  002110  010146                           MOV     R1,-(SP)
1130  002112  012701  000004                   MOV     #ILLMEM,R1
1131  002116  012721  013742                   MOV     #INTSRV,(R1)+
1132  002122  012711  000340                   MOV     #PR7,(R1)
1133  002126  012601                           MOV     (SP)+,R1
1134  002130                                                                   LET     I := #0
1135  002130  005037  001270                   CLR     I
1136  002134                                                                   REPEAT
1137  002134                           $3:
1138  002134                                                                       BGNSUB
1139  002134  012737  002142  001110           MOV     #64$,$LPERR
1140                                                                                 ;CLEAR FLAG
1141  002142                                                                       LET INTFLAG := #0
1142  002142  005037  013750                   CLR     INTFLAG
1143
1144                                                                                 ;READ FLAG
1145  002146  005711                           TST     @ADRS
1146  002150                                                                       IF INTFLAG NE #0 THEN
1147  002150  005737  013750                   TST     INTFLAG
1148  002154  001401                           BEQ     $4
1149                                                                                 ; FATAL ERROR
1150  002156                                                                           ERRDF 1
1151  002156  104001                           ERROR   1
1152  002160                                                                       ENDIF
1153  002160                           $4:
1154  002160                                                                   ENDSUB
1155  002160                                                                   LET     I := I + #2
1156  002160  062737  000002  001270           ADD     #2,I
1157  002166                                                                   LET     ADRS := DLADD + I
1158  002166  013701  001254                   MOV     DLADD,ADRS
1159  002172  063701  001270                   ADD     I,ADRS
1160  002176                                                           UNTIL I EQ #8.
1161  002176  023727  001270  000010           CMP     I,#8.
1162  002204  001353                           BNE     $3
```

```
1163
1164  002206  032777  177476  177044          BIT     #177476,@RCSR   ;CHECK THAT ALL UNUSED BITS ARE 0
1165  002214  001401                          BEQ     1$
1166  002216  104027                          ERROR   27              ;RCSR HAS UNUSED BITS SET
1167
1168  002220  032777  007400  177034  1$:     BIT     #7400,@RBUF
1169  002226  001401                          BEQ     2$
1170  002230  104030                          ERROR   30              ;RBUF HAS UNUSED BITS SET
1171
1172  002232  032777  177476  177024  2$:     BIT     #177476,@TCSR
1173  002240  001401                          BEQ     3$
1174  002242  104031                          ERROR   31              ;TCSR HAS UNUSED BITS SET
1175
1176  002244  032777  177400  177014  3$:     BIT     #177400,@TBUF
1177  002252  001401                          BEQ     4$
1178  002254  104032                          ERROR   32              ;TBUF HAS ;UNUSED BITS SET
1179
1180  002256                          4$:                                     CLRVEC ILLMEM
1181  002256  010146                          MOV     R1,-(SP)
1182  002260  010246                          MOV     R2,-(SP)
1183  002262  012701  000004                  MOV     #ILLMEM,R1
1184  002266  010102                          MOV     R1,R2
1185  002270  062702  000002                  ADD     #2,R2
1186  002274  010221                          MOV     R2,(R1)+
1187  002276  005011                          CLR     (R1)
1188  002300  012602                          MOV     (SP)+,R2
1189  002302  012601                          MOV     (SP)+,R1
1190                                                                  ;END OF TEST
```

```
                                ;;*****************************************************************
                                ;* THE FOLLOWING 3 TESTS  TEST ALL 'READ WRITE' BITS
                                ;;*****************************************************************


                                ;;*****************************************************************
                                ;*TEST 2        BREAK - TCSR 0 SET, CLEAR, RESET
                                ;*               THIS BIT IS THE ONLY ONE IN THIS POSITION
                                ;*               THAT IS READ AND WRITE.
                                ;;*****************************************************************
002304  000004                  TST2:   SCOPE
002306  012737  000010  001160          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
002314  012737  000002  001200          MOV     #2,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX

002322                                                                  IF #BRK NOTSETIN $USWR THEN
002322  032737  000010  001220          BIT     #BRK,$USWR
002330  001001                        BNE     $5
002332                                                                      EXIT                ; BREAK NOT INSTALLED
002332  000452                        BR      TST3            ;;EXIT THIS TEST
002334                                                                  ENDIF
002334                          $5:

                                                ; SEE IF IT IS CLEAR
002334                                                                  BGNSUB
002334  012737  002342  001110          MOV     #64$,$LPERR

002342                                                                      IF      #BREAK SETIN @TCSR THEN
002342  032777  000001  176714          BIT     #BREAK,@TCSR
002350  001401                        BEQ     $6

                                                                            ; BREAK DID NOT RESET IN TCSR
002352                                                                      ERRHRD 2
002352  104002                        ERROR   2
002354                                                                  ENDIF
002354                          $6:
002354                                                              ENDSUB

                                                    ; TRY TO SET BREAK BIT
002354                                                              BGNSUB
002354  012737  002362  001110          MOV     #64$,$LPERR
002362                                                                  LET     @TCSR := @TCSR SET.BY #BREAK
002362  052777  000001  176674          BIS     #BREAK,@TCSR

                                                                        ; STUCK TO 0
002370                                                                  IF      #BREAK NOTSETIN @TCSR THEN
002370  032777  000001  176666          BIT     #BREAK,@TCSR
002376  001001                        BNE     $7

                                                                        ; BREAK DID NOT SET IN TCSR
002400                                                                  ERRHRD 3
002400  104003                        ERROR   3
002402                                                                  ENDIF
002402                          $7:
002402                                                              ENDSUB

                                                    ; TRY TO CLEAR A SET BIT
002402                                                              BGNSUB
002402  012737  002410  001110          MOV     #64$,$LPERR
```

```
1247  002410
1248  002410  042777  000001  176646          BIC     #BREAK,@TCSR
1249
1250  002416
1251  002416  032777  000001  176640          BIT     #BREAK,@TCSR
1252  002424  001401                          BEQ     $10
1253
1254  002426
1255  002426  104004                          ERROR   4
1256  002430
1257  002430                          $10:
1258  002430
1259
1260                                          ; NOW SEE IF RESET CLEARS IT
1261  002430
1262  002430  012737  002436  001110          MOV     #64$,$LPERR
1263
1264  002436
1265  002436  052777  000001  176620          BIS     #BREAK,@TCSR
1266
1267  002444
1268  002444  000005                          RESET
1269  002446
1270  002446  032777  000001  176610          BIT     #BREAK,@TCSR
1271  002454  001401                          BEQ     $11
1272
1273  002456
1274  002456  104005                          ERROR   5
1275  002460
1276  002460                          $11:
1277  002460
```

```
                LET     @TCSR := @TCSR CLR.BY #BREAK

                    ; SHOULD HAVE CLEARED
                IF      #BREAK SETIN @TCSR THEN

                    ; BREAK  DID NOT CLEAR IN TCSR
                    ERRHRD 4

                ENDIF

            ENDSUB

                    ; NOW SEE IF RESET CLEARS IT
            BGNSUB

                LET     @TCSR := @TCSR SET.BY #BREAK

                    ; ISSUE BUS RESET
                BRESET

                IF      #BREAK SETIN @TCSR THEN

                    ; BREAK DID NOT RESET IN TCSR
                    ERRHRD 5

                ENDIF

            ENDSUB
```

```
                                ;;*****************************************************************
                                ;*TEST 3          XMITIE - TCSR 6          SET, CLEAR, RESET
                                ;;*****************************************************************
002460  000004                  TST3:   SCOPE
002462  012737  000010  001160          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
002470  012737  000003  001200          MOV     #3,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
                                                                ; USE PRIORITY OF 7
002476                                  SETPRI #PR7

                                                ; SEE IF IT IS CLEAR
                                                                BGNSUB
002510  012737  002516  001110          MOV     #65$,$LPERR

                                                                    IF      #XMITIE SETIN @TCSR THEN
002516  032777  000100  176540          BIT     #XMITIE,@TCSR
002524  001401                          BEQ     $12

                                                                    ; XMITIE DID NOT RESET IN TCSR
                                                                    ERRHRD 12
002526  104012                          ERROR   12
                                                                    ENDIF
002530                          $12:
                                                                ENDSUB

                                                ; TRY TO SET XMITIE BIT
                                                                BGNSUB
002530  012737  002536  001110          MOV     #64$,$LPERR
                                                                    LET     @TCSR := @TCSR SET.BY #XMITIE
002536  052777  000100  176520          BIS     #XMITIE,@TCSR

                                                                    ; STUCK TO 0
                                                                    IF      #XMITIE NOTSETIN @TCSR THEN
002544  032777  000100  176512          BIT     #XMITIE,@TCSR
002552  001001                          BNE     $13

                                                                    ; XMIT DID NOT SET IN TCSR
                                                                    ERRHRD 13
002554  104013                          ERROR   13
                                                                    ENDIF
002556                          $13:
                                                                ENDSUB

                                                ; TRY TO CLEAR A SET BIT
                                                                BGNSUB
002556  012737  002564  001110          MOV     #64$,$LPERR

                                                                    LET     @TCSR := @TCSR CLR.BY #XMITIE
002564  042777  000100  176472          BIC     #XMITIE,@TCSR

                                                                    ; SHOULD HAVE CLEARED
                                                                    IF      #XMITIE SETIN @TCSR THEN
002572  032777  000100  176464          BIT     #XMITIE,@TCSR
002600  001401                          BEQ     $14

                                                                    ;  XMIT DID NOT CLEAR IN TCSR
                                                                    ERRHRD 14
002602  104014                          ERROR   14
                                                                    ENDIF
002604                          $14:
                                                                ENDSUB
002604
```

```
1334
1335                                                    ; NOW SEE IF RESET CLEARS IT
1336  002604                                                  BGNSUB
1337  002604  012737  002612  001110          MOV     #64$,$LPERR
1338
1339  002612                                                      LET     @TCSR := @TCSR SET.BY #XMITIE
1340  002612  052777  000100  176444          BIS     #XMITIE,@TCSR
1341                                                          ; ISSUE BUS RESET
1342  002620                                                  BRESET
1343  002620  000005                          RESET
1344  002622                                                  IF      #XMITIE SETIN @TCSR THEN
1345  002622  032777  000100  176434          BIT     #XMITIE,@TCSR
1346  002630  001401                          BEQ     $15
1347                                                          ; XMIT DID NOT RESET IN TCSR
1348  002632                                                  ERRHRD 15
1349  002632  104015                          ERROR   15
1350  002634                                                  ENDIF
1351  002634                          $15:
1352  002634                                                  ENDSUB
```

```
                                ;;*****************************************************************
                                ;*TEST 4          RCVRIE - RCSR 6         SET, CLEAR, RESET
                                ;*               THIS BIT IS THE ONLY ONE IN THIS POSITION
                                ;*               THAT IS READ AND WRITE.
                                ;;*****************************************************************
002634  000004                  TST4:   SCOPE
002636  012737  000010  001160          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
002644  012737  000004  001200          MOV     #4,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
                                                        ; SEE IF IT IS CLEAR
002652                                                          BGNSUB
002652  012737  002660  001110          MOV     #64$,$LPERR
                                                                IF      #RCVRIE SETIN @RCSR THEN
002660  032777  000100  176372          BIT     #RCVRIE,@RCSR
002666  001401                        BEQ     $16
                                                                    ; RCVRIE DID NOT RESET IN RCSR
                                                                    ERRHRD 35
002670  104035                        ERROR   35
                                                                ENDIF
002672                          $16:
002672                                                          ENDSUB
                                                        ; TRY TO SET RCVRIE BIT
002672                                                          BGNSUB
002672  012737  002700  001110          MOV     #64$,$LPERR
002700                                                          LET     @RCSR := @RCSR SET.BY #RCVRIE
002700  052777  000100  176352          BIS     #RCVRIE,@RCSR
                                                                ; STUCK TO 0
002706                                                          IF      #RCVRIE NOTSETIN @RCSR THEN
002706  032777  000100  176344          BIT     #RCVRIE,@RCSR
002714  001001                        BNE     $17
                                                                    ; RCVRIE DID NOT SET IN RCSR
                                                                    ERRHRD 36
002716  104036                        ERROR   36
                                                                ENDIF
002720                          $17:
002720                                                          ENDSUB
                                                        ; TRY TO CLEAR A SET BIT
002720                                                          BGNSUB
002720  012737  002726  001110          MOV     #64$,$LPERR
                                                                LET     @RCSR := @RCSR CLR.BY #RCVRIE
002726  042777  000100  176324          BIC     #RCVRIE,@RCSR
                                                                ; SHOULD HAVE CLEARED
002734                                                          IF      #RCVRIE SETIN @RCSR THEN
002734  032777  000100  176316          BIT     #RCVRIE,@RCSR
002742  001401                        BEQ     $20
                                                                    ; RCVRIE DID NOT CLEAR IN RCSR
                                                                    ERRHRD 37
002744  104037                        ERROR   37
                                                                ENDIF
002746                          $20:
002746                                                          ENDSUB
```

```
1409                                                    ; NOW SEE IF RESET CLEARS IT
1410  002746                                                  BGNSUB
1411  002746  012737  002754  001110          MOV     #64$,$LPERR
1412
1413  002754                                                    LET     @RCSR := @RCSR SET.BY #RCVRIE
1414  002754  052777  000100  176276          BIS     #RCVRIE,@RCSR
1415                                                        ; ISSUE BUS RESET
1416  002762                                                  BRESET
1417  002762  000005                  RESET
1418  002764                                                    IF      #RCVRIE SETIN @RCSR THEN
1419  002764  032777  000100  176266          BIT     #RCVRIE,@RCSR
1420  002772  001401                  BEQ     $21
1421                                                        ; RCVRIE DID NOT RESET IN RCSR
1422  002774                                                  ERRHRD 40
1423  002774  104040                  ERROR   40
1424  002776                                                    ENDIF
1425  002776                  $21:
1426  002776                                                  ENDSUB
```

```
1427
1428                                    ;;*****************************************************************
1429                                    ;*TEST 5          TEST THAT XMITRDY - TCSR 7 - IS SET BY INIT
1430                                    ;;*****************************************************************
1431  002776  000004                    TST5:   SCOPE
1432  003000  012737  000010  001160            MOV     #10,$TIMES      ;;DO 10 ITERATIONS
1433  003006  012737  000005  001200            MOV     #5,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
1434
1435  003014                                                                  BGNSUB
1436  003014  012737  003022  001110            MOV     #64$,$LPERR
1437
1438  003022                                                                      IF      #XMITRDY NOTSETIN @TCSR THEN
1439  003022  032777  000200  176234            BIT     #XMITRDY,@TCSR
1440  003030  001002                            BNE     $22
1441
1442                                                                                ;RESET SHOULD HAVE SET BIT.
1443                                                                                ;XMITRDY DID NOT SET IN TCSR (AFTER RESE
1444  003032                                                                        ERRHRD 42
1445  003032  104042                            ERROR   42
1446                                                                                ;ISSUE ANOTHER RESET
1447  003034                                                                        BRESET
1448  003034  000005                            RESET
1449  003036                                                                    ENDIF
1450  003036                                $22:
1451                                                                                ;ALLOW LOOPING ON ERROR
1452  003036                                                                  ENDSUB
```

```
 1453
 1454                                ;;*****************************************************************
 1455                                ;*TEST 6         TEST THAT XMIT RDY - TCSR 7 - CLEARS
 1456                                ;*              WHEN TBUF IS LOADED WITH A CHARACTER
 1457                                ;*              AND THAT IT SETS WITHIN A REASONABLE AMOUNT OF TIME.
 1458                                ;;*****************************************************************
 1459  003036  000004                TST6:   SCOPE
 1460  003040  012737  000010  001160        MOV     #10,$TIMES      ;;DO 10 ITERATIONS
 1461  003046  012737  000006  001200        MOV     #6,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
 1462
 1463  003054                                                        IF CONSOL EQ #TRUE THEN
 1464  003054  023727  013736  000001        CMP     CONSOL,#TRUE
 1465  003062  001001                        BNE     $23
 1466  003064                                                           EXIT
 1467  003064  000521                        BR      TST7            ;;EXIT THIS TEST
 1468  003066                                                        ENDIF
 1469  003066                        $23:
 1470
 1471  003066                                                        LET PASS := #0  ;INIT COUNT OF TIMES THRU
 1472  003066  005037  003322                CLR     PASS
 1473  003072                                                        LOOP            ; START OF LOOP
 1474  003072                        $24:
 1475                                                                ;  MAX OF 2 TIMES THRU
 1476  003072                                                           LET ERRORFLAG := #0
 1477  003072  005037  003324                CLR     ERRORFLAG
 1478  003076                                                            LET EXITFLAG := #0
 1479  003076  005037  003326                CLR     EXITFLAG
 1480                                                    ; LOAD TBUF WITH ONE CHARACTER
 1481                                                    ; WAIT FOR READY TO SET
 1482                                                    ; (SHOULD BE VERY SHORT WAIT
 1483                                                    ; SINCE UART DOUBLE BUFFERS ITS INPUT)
 1484
 1485                                                        ;SEND A CHARACTER
 1486  003102                                                LET @TBUF :B= #0
 1487  003102  105077  176160                CLRB    @TBUF
 1488
 1489                                                        ;WAIT A MAXIMUM
 1490                                                        ;OF 50 MSEC FOR
 1491  003106                                                ;XMIT RDY TO SET IN TCSR
 1492  003106  010546                        MOV     R5,-(SP)        CALL TIMER IN <#500,#XMITRDY,TCSR,#SET>
 1493  003110  012745  177777                MOV     #SET,-(R5)
 1494  003114  013745  001264                MOV     TCSR,-(R5)
 1495  003120  012745  000200                MOV     #XMITRDY,-(R5)
 1496  003124  012745  000500                MOV     #500,-(R5)
 1497  003130  004737  012560                JSR     PC,TIMER
 1498  003134  012605                        MOV     (SP)+,R5
 1499
 1500                                                        ;TIMER RETURNS AN ERROR IF BIT DID
 1501  003136                                                ;NOT MEET CONDITION WITHIN TIME LIMIT
 1502  003136  103003                        BCC     $26     IF.ERROR THEN
 1503
 1504  003140                                                ;XMIT RDY DID NOT SET IN TCSR
 1505  003140  104066                        ERROR   66      ERRHRD 66
 1506  003142  000137  003330                JMP     TST7
 1507  003146                                                ENDIF
 1508  003146                        $26:
```

```
 1509  003146  105077  176114          CLRB    @TBUF   ;SHIP 1'ST CHAR
 1510  003152  105777  176106  1$:     TSTB    @TCSR   ;WAIT FOR RDY
 1511  003156  100375                  BPL     1$
 1512                                      ; LOAD TBUF WITH A SECOND CHARACTER (TO DOUBLE BUFFER)
 1513                                      ; CHECK IMMEDIATELY THAT XMITRDY IS CLEAR
 1514                                      ; AND THEN WAIT FOR IT TO SET
 1515
 1516                                                          ;SEND SECOND CHARACTER
 1517  003160                                                LET @TBUF :B= #0
 1518  003160  105077  176102          CLRB    @TBUF
 1519  003164  000240                          NOP             ; GIVE IT TIME TO CLEAR
 1520                                                          ; XMITRDY SHOULD HAVE CLEARED UPON
 1521                                                          ; RECEIPT OF A CHARACTER
 1522  003166                                                IF #XMITRDY SETIN @TCSR THEN
 1523  003166  032777  000200  176070  BIT     #XMITRDY,@TCSR
 1524  003174  001404                  BEQ     $27
 1525                                                          ; XMITRDY DID NOT CLEAR IN TCSR
 1526  003176                                                  LET ERRORFLAG := #SET
 1527  003176  012737  177777  003324  MOV     #SET,ERRORFLAG
 1528                                                                  ; DEFER ERROR TYPEOUT
 1529
 1530  003204                                                ELSE
 1531  003204  000416                  BR      $30
 1532  003206                  $27:
 1533                                                          ;WAIT A MAXIMUM
 1534                                                          ;OF 100 MSEC FOR
 1535                                                          ;XMIT RDY TO SET IN TCSR
 1536  003206                                                CALL TIMER IN <#1000,#XMITRDY,TCSR,#SET>
 1537  003206  010546                  MOV     R5,-(SP)
 1538  003210  012745  177777          MOV     #SET,-(R5)
 1539  003214  013745  001264          MOV     TCSR,-(R5)
 1540  003220  012745  000200          MOV     #XMITRDY,-(R5)
 1541  003224  012745  001000          MOV     #1000,-(R5)
 1542  003230  004737  012560          JSR     PC,TIMER
 1543  003234  012605                  MOV     (SP)+,R5
 1544  003236
 1545  003236  103001                  BCC     $31                 IF.ERROR THEN
 1546
 1547  003240                                                          ;XMIT RDY DID NOT SET IN TCSR
 1548  003240  104070                  ERROR   70                      ERRHRD 70
 1549  003242                                                      ENDIF
 1550  003242                  $31:                              ENDIF ; OF DEFERED ERROR CALL
 1551  003242
 1552  003242                  $30:                          IF ERRORFLAG EQ #SET THEN
 1553  003242
 1554  003242  023727  003324  177777  CMP     ERRORFLAG,#SET
 1555  003250  001013                  BNE     $32             LET PASS := PASS + #1
 1556  003252
 1557  003252  005237  003322          INC     PASS              IF PASS GT #1 THEN
 1558  003256
 1559  003256  023727  003322  000001  CMP     PASS,#1
 1560  003264  003404                  BLE     $33                   ; CALL ERROR IF 2ND TRY
 1561                                                                 ERRHRD 67
 1562  003266
 1563  003266  104067                  ERROR   67                     LET EXITFLAG := #SET
 1564  003270
```

```
1565  003270  012737  177777  003326          MOV     #SET,EXITFLAG
1566  003276                                                              ENDIF
1567  003276                          $33:
1568  003276                                                          ELSE            ; NO ERROR
1569  003276  000403                         BR      $34
1570  003300                          $32:
1571  003300                                                              LET EXITFLAG := #SET
1572  003300  012737  177777  003326          MOV     #SET,EXITFLAG
1573  003306                                                          ENDIF
1574  003306                          $34:
1575  003306                                                      EXIF        EXITFLAG EQ #SET
1576  003306  023727  003326  177777          CMP     EXITFLAG,#SET
1577  003314  001401                         BEQ     $25
1578  003316                                                      ENDLOOP
1579  003316  000665                         BR      $24
1580  003320                          $25:
1581  003320                                                          EXIT ; SKIP AROUND FLAG WORDS
1582  003320  000403                         BR      TST7            ;;EXIT THIS TEST
1583  003322  000000                  PASS:       0
1584  003324  000000                  ERRORFLAG:  0
1585  003326  000000                  EXITFLAG:   0
```

```
                                ;;*****************************************************************
                                ;*TEST 7         TEST THAT OUTPUTTING A CHAR FROM TBUF (WITH WRAP CONNECTED)
                                ;*              RESULTS IN RCVRDONE SETTING WITHIN A REASONABLE AMOUNT OF TIME
                                ;*              AND THAT RESET CLEARS THE BIT.
                                ;;*****************************************************************
003330  000004                TST7:   SCOPE
003332  012737  000010  001160        MOV     #10,$TIMES      ;;DO 10 ITERATIONS
003340  012737  000007  001200        MOV     #7,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX

003346                                                          IF #WRAP NOTSETIN $USWR OR CONSOL EQ #TRUE THEN
003346  032737  000020  001220        BIT     #WRAP,$USWR
003354  001404                        BEQ     $35
003356  023727  013736  000001        CMP     CONSOL,#TRUE
003364  001001                        BNE     $36
003366                          $35:
003366                                                              EXIT
003366  000434                        BR      TST10           ;;EXIT THIS TEST
003370                                                          ENDIF
003370                          $36:

003370                                                          BGNSUB
003370  012737  003376  001110        MOV     #64$,$LPERR
                                        ; SEND A CHARACTER AND LET IT WRAP AROUND

003376                                                              LET @TBUF :B= #0
003376  105077  175604                CLRB    @TBUF

                                                                ; WAIT A MAXIMUM OF 50 MSEC
                                                                ; FOR RCVR DONE TO SET IN
                                                                ; RCSR
003402                                                          CALL TIMER IN <#500,#RCVRDONE,RCSR,#SET>
003402  010546                        MOV     R5,-(SP)
003404  012745  177777                MOV     #SET,-(R5)
003410  013745  001260                MOV     RCSR,-(R5)
003414  012745  000200                MOV     #RCVRDONE,-(R5)
003420  012745  000500                MOV     #500,-(R5)
003424  004737  012560                JSR     PC,TIMER
003430  012605                        MOV     (SP)+,R5

                                                                ;DIDN'T SET IN TIME
003432                                                          IF.ERROR THEN
003432  103001                        BCC     $37

                                                                ; RCVRDONE DID NOT SET IN RCSR
                                                                ERRHRD 71
003434
003434  104071                        ERROR   71
003436                                                          ENDIF
003436                          $37:


003436                                                      ENDSUB

003436                                                      BGNSUB
003436  012737  003444  001110        MOV     #64$,$LPERR
                                        ; NOW THAT IT IS SET SEE IF IT CAN BE RESET
003444                                                          BRESET
003444  000005                        RESET
```

```
1642
1643  003446                                                    IF #RCVRDONE SETIN @RCSR THEN
1644  003446  032777  000200  175604          BIT     #RCVRDONE,@RCSR
1645  003454  001401                          BEQ     $40
1646                                                        ; RCVRDONE DID NOT RESET IN RCSR.
1647  003456                                                    ERRHRD 72
1648  003456  104072                          ERROR   72
1649  003460                                                    ENDIF
1650  003460                          $40:
1651  003460                                                ENDSUB
```

```
1652
1653                                    ;;*********************************************************************
1654                                    ;*TEST 10        TEST THAT RCVRDONE IS CLEARED BY SETTING READER ENABLE
1655                                    ;;*********************************************************************
1656  003460  000004                    TST10:  SCOPE
1657  003462  012737  000010  001160            MOV     #10,$TIMES      ;;DO 10 ITERATIONS
1658  003470  012737  000010  001200            MOV     #10,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1659
1660  003476                                                            IF #WRAP NOTSETIN $USWR OR CONSOL EQ #TRUE THEN
1661  003476  032737  000020  001220            BIT     #WRAP,$USWR
1662  003504  001404                           BEQ     $41
1663  003506  023727  013736  000001            CMP     CONSOL,#TRUE
1664  003514  001001                           BNE     $42
1665  003516                            $41:
1666  003516                                                                EXIT
1667  003516  000433                           BR      TST11           ;;EXIT THIS TEST
1668  003520                                                            ENDIF
1669  003520                            $42:
1670
1671  003520                                                            BGNSUB
1672  003520  012737  003526  001110            MOV     #64$,$LPERR
1673                                                  ; OUTPUT A CHARACTER AND WAIT FOR XMITRDY TO SET.
1674
1675                                                                        ; OUTPUT A CHARACTER
1676  003526                                                              LET @TBUF :B= #0
1677  003526  105077  175534                   CLRB    @TBUF
1678                                                                      ; WAIT MAXIMUM OF 500 MSEC
1679                                                                      ; FOR RCVRDONE TO SET IN
1680                                                                      ; RCSR
1681  003532                                                              CALL TIMER IN <#500,#RCVRDONE,RCSR,#SET>
1682  003532  010546                           MOV     R5,-(SP)
1683  003534  012745  177777                   MOV     #SET,-(R5)
1684  003540  013745  001260                   MOV     RCSR,-(R5)
1685  003544  012745  000200                   MOV     #RCVRDONE,-(R5)
1686  003550  012745  000500                   MOV     #500,-(R5)
1687  003554  004737  012560                   JSR     PC,TIMER
1688  003560  012605                           MOV     (SP)+,R5
1689
1690  003562                                                                  ; DID IT BECOME READY?
1691  003562  103001                           BCC     $43                   IF.ERROR  THEN
1692
1693  003564                                                                  ;RCVRDONE DID NOT SET IN RCSR
1694  003564  104025                           ERROR   25                      ERRHRD 25
1695
1696  003566                                                                  ; SET IT BACK TO CONTINUE
1697  003566                            $43:                                ENDIF
1698  003566                                                            ENDSUB
1699
1700                                                  ;NOW THAT IT IS SET LETS SEE IF SETTING
1701                                                  ;READER ENABLE CLEARS RCVRDONE
1702
1703  003566                                                            LET @RCSR := @RCSR SET.BY #RDRRUN
1704  003566  052777  000001  175464            BIS     #RDRRUN,@RCSR
1705  003574                                                                IF #RCVRDONE SETIN @RCSR THEN
1706  003574  032777  000200  175456            BIT     #RCVRDONE,@RCSR
1707  003602  001401                           BEQ     $44
```

```
1708                                                                    ;RCVRDONE DID NOT CLEAR IN RCSR
1709  003604                                                            ERRHRD 26
1710  003604  104026                      ERROR   26
1711                                                                    ; SET IT BACK TO CONTINUE
1712  003606                                                          ENDIF
1713  003606                      $44:
```

```
1714
1715                                    ;;*****************************************************************
1716                                    ;*TEST 11        TEST THAT RCVRDONE IS CLEARED BY READING RBUF
1717                                    ;;*****************************************************************
1718  003606  000004                    TST11:  SCOPE
1719  003610  012737  000010  001160            MOV     #10,$TIMES      ;;DO 10 ITERATIONS
1720  003616  012737  000011  001200            MOV     #11,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1721
1722  003624                                                            IF #WRAP NOTSETIN $USWR OR CONSOL EQ #TRUE THEN
1723  003624  032737  000020  001220            BIT     #WRAP,$USWR
1724  003632  001404                           BEQ     $45
1725  003634  023727  013736  000001            CMP     CONSOL,#TRUE
1726  003642  001001                           BNE     $46
1727  003644                            $45:                                EXIT
1728  003644
1729  003644  000432                           BR      TST12           ;;EXIT THIS TEST
1730  003646                                                            ENDIF
1731  003646                            $46:
1732
1733  003646                                                            BGNSUB
1734  003646  012737  003654  001110            MOV     #64$,$LPERR
1735                                                  ; OUTPUT A CHARACTER AND WAIT FOR RCVRDONE TO SET.
1736
1737                                                                        ; OUTPUT A CHARACTER
1738  003654                                                            LET @TBUF :B= #0
1739  003654  105077  175406                   CLRB    @TBUF
1740                                                                        ; WAIT MAXIMUM OF 500 MSEC
1741                                                                        ; FOR RCVRDONE TO SET IN
1742                                                                        ; RCSR
1743  003660                                                            CALL TIMER IN <#500,#RCVRDONE,RCSR,#SET>
1744  003660  010546                           MOV     R5,-(SP)
1745  003662  012745  177777                   MOV     #SET,-(R5)
1746  003666  013745  001260                   MOV     RCSR,-(R5)
1747  003672  012745  000200                   MOV     #RCVRDONE,-(R5)
1748  003676  012745  000500                   MOV     #500,-(R5)
1749  003702  004737  012560                   JSR     PC,TIMER
1750  003706  012605                           MOV     (SP)+,R5
1751                                                                        ; DID IT BECOME READY?
1752  003710                                                            IF.ERROR  THEN
1753  003710  103001                           BCC     $47
1754                                                                        ;RCVRDONE DID NOT SET IN RCSR
1755  003712                                                                ERRHRD 73
1756  003712  104073                           ERROR   73
1757                                                                        ; SET IT BACK TO CONTINUE
1758  003714                                                            ENDIF
1759  003714                            $47:
1760  003714                                                    ENDSUB
1761
1762                                                  ; NOW THAT IT IS SET LETS SEE IF READING THE
1763                                                  ; BUFFER CLEARS RCVRDONE.
1764
1765                                                                        ;READ BUFFER
1766  003714                                                            LET R0 :B= @RBUF
1767  003714  117700  175342                   MOVB    @RBUF,R0
1768
1769  003720                                                            IF #RCVRDONE SETIN @RCSR THEN
```

```
 1770  003720  032777  000200  175332          BIT     #RCVRDONE,@RCSR
 1771  003726  001401                          BEQ     $50
 1772                                                                    ;RCVRDONE DID NOT CLEAR IN RCSR
 1773  003730                                                            ERRHRD 74
 1774  003730  104074                          ERROR   74
 1775                                                                    ; SET IT BACK TO CONTINUE
 1776  003732                                                          ENDIF
 1777  003732                        $50:
```

```
                                ;;*****************************************************************
                                ;*TEST 12       TEST THE OVERRUN & ERROR BITS - RBUF 14
                                ;;*****************************************************************
003732  000004          TST12:  SCOPE
003734  012737  000010  001160          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
003742  012737  000012  001200          MOV     #12,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX

003750                                                                  IF CONSOL EQ #0 THEN
003750  005737  013736                  TST     CONSOL
003754  001007                          BNE     $51
003756                                                                      IF #WRAP SETIN $USWR THEN
003756  032737  000020  001220          BIT     #WRAP,$USWR
003764  001401                          BEQ     $52
                                                                                ;; NULL ---EXECUTE TEST
003766                                                                      ELSE
003766  000401                          BR      $53
003770                          $52:
003770                                                                          EXIT
003770  000524                          BR      TST13           ;;EXIT THIS TEST
003772                                                                      ENDIF
003772                          $53:
003772                                                                  ELSE
003772  000401                          BR      $54
003774                          $51:
003774                                                                      EXIT
003774  000522                          BR      TST13           ;;EXIT THIS TEST
003776                                                                  ENDIF
003776                          $54:

                                .
                                ;
                                ;

003776                                                                  BGNSUB
003776  012737  004004  001110          MOV     #64$,$LPERR
                                            ;OUTPUT 2 CHARACTERS
                                            ;THIS SHOULD AN CAUSE OVERRUN ERROR.

                                                                            ;OUTPUT 1 CHARACTER
004004                                                                  LET @TBUF :B= #0
004004  105077  175256                  CLRB    @TBUF

004010                                                                  CALL TIMER IN <#500,#RCVRDONE,RCSR,#SET>
004010  010546                          MOV     R5,-(SP)
004012  012745  177777                  MOV     #SET,-(R5)
004016  013745  001260                  MOV     RCSR,-(R5)
004022  012745  000200                  MOV     #RCVRDONE,-(R5)
004026  012745  000500                  MOV     #500,-(R5)
004032  004737  012560                  JSR     PC,TIMER
004036  012605                          MOV     (SP)+,R5

                                                                            ;DID IT SET IN TIME?
004040                                                                  IF.ERROR THEN
004040  103001                          BCC     $55

                                                                            ;RCVRDONE DID NOT SET IN RCSR
                                                                            ERRHRD 16
004042  104016                          ERROR   16
```

```
                                                                ENDIF
                                $55:
                                                                    ;OUTPUT 2ND CHARACTER
                                                                LET @TBUF :B= #0
004044  105077  175216          CLRB    @TBUF
                                                                    ;LET OVERRUN HAPPEN
                                                                    WAITMS 300.
004050
004050  010546                  MOV     R5,-(SP)
004052  012745  000454          MOV     #300.,-(R5)
004056  004737  012732          JSR     PC,WAIT
004062  012605                  MOV     (SP)+,R5

                                                                    ;READ BUFFER AND ERROR BITS
004064                                                          LET R4 := @RBUF
004064  017704  175172          MOV     @RBUF,R4

                                                                    ;IT DIDN'T SET
004070                                                          IF #ORERR NOTSETIN R4 THEN
004070  032704  040000          BIT     #ORERR,R4
004074  001002                  BNE     $56
                                                                    ;ORERR DID NOT SET IN RBUF
                                                                    ERRHRD 101
004076
004076  104101                  ERROR   101

                                                                    ;NO USE COMPOUNDING ERRORS
                                                                    EXIT
004100
004100  000460                  BR      TST13           ;;EXIT THIS TEST
004102                                                              ENDIF
                                $56:
                                                                ENDSUB

                                ;NOW SEE IF ERROR BIT SET WITH OVERRUN  ERROR:
004102                                                          BGNSUB
004102  012737  004110  001110  MOV     #64$,$LPERR
004110                                                              IF #ERROR NOTSETIN R4 THEN
004110  032704  100000          BIT     #ERROR,R4
004114  001002                  BNE     $57

                                                                    ;ERROR DID NOT SET IN RBUF
                                                                    ERRHRD 102
004116
004116  104102                  ERROR   102

                                                                    ;-WHEN ORERR SET.
                                                                    ;GET OUT NOW.
                                                                    EXIT
004120
004120  000450                  BR      TST13           ;;EXIT THIS TEST
004122                                                              ENDIF
                                $57:
                                                                ENDSUB

                                                                BGNSUB
004122  012737  004130  001110  MOV     #64$,$LPERR
                                    ;CHECK REAL RBUF TO SEE IF ORERR IS STILL SET.

                                                                    IF #ORERR NOTSETIN @RBUF THEN
```

```
1890  004130  032777  040000  175124          BIT     #ORERR,@RBUF
1891  004136  001002                          BNE     $60
1892
1893                                                                  ;READING RBUF CLEARED ORERR.
1894  004140                                                          ERRHRD 103
1895  004140  104103                          ERROR   103
1896                                                                  ;SKIP REST OF TEST
1897  004142                                                          EXIT
1898  004142  000437                          BR      TST13          ;;EXIT THIS TEST
1899  004144                                                              ENDIF
1900  004144                          $60:
1901  004144                                                      ENDSUB
1902
1903  004144                                                      BGNSUB
1904  004144  012737  004152  001110          MOV     #64$,$LPERR
1905                                              ;READING RBUF ABOVE SHOULD ENABLE ERROR TO BE CLEARED
1906                                              ;BY NEXT TRANSFER.
1907                                              ;NOW SEE IF THEY CLEAR WHEN ANOTHER CHAR. IS RECEIVED
1908
1909                                                                  ;SEND A CHARACTER AROUND.
1910  004152                                                      LET @TBUF :B= #0
1911  004152  105077  175110                  CLRB    @TBUF
1912  004156                                                      CALL TIMER IN <#500,#RCVRDONE,RCSR,#SET>
1913  004156  010546                          MOV     R5,-(SP)
1914  004160  012745  177777                  MOV     #SET,-(R5)
1915  004164  013745  001260                  MOV     RCSR,-(R5)
1916  004170  012745  000200                  MOV     #RCVRDONE,-(R5)
1917  004174  012745  000500                  MOV     #500,-(R5)
1918  004200  004737  012560                  JSR     PC,TIMER
1919  004204  012605                          MOV     (SP)+,R5                    ``
1920
1921                                                                  ;DID IT SET IN TIME?
1922  004206                                                      IF.ERROR THEN
1923  004206  103001                          BCC     $61
1924                                                                  ;RCVRDONE DID NOT SET IN RCSR
1925  004210                                                          ERRHRD 20
1926  004210  104020                          ERROR   20
1927  004212                                                      ENDIF
1928  004212                          $61:
1929
1930  004212                                                          IF #ORERR SETIN @RBUF THEN
1931  004212  032777  040000  175042          BIT     #ORERR,@RBUF
1932  004220  001402                          BEQ     $62
1933                                                                  ;ORERR DID NOT CLEAR IN RBUF
1934  004222                                                          ERRHRD 104
1935  004222  104104                          ERROR   104
1936
1937                                                                  ;-AFTER RECEIVING ANOTHER CHAR
1938                                                                  ;SKIP AROUND REST
1939  004224                                                          EXIT
1940  004224  000406                          BR      TST13          ;;EXIT THIS TEST
1941  004226                                                              ENDIF
1942  004226                          $62:
1943
1944  004226                                                          IF #ERROR SETIN @RBUF THEN
1945  004226  032777  100000  175026          BIT     #ERROR,@RBUF
```

```
 1946  004234  001401                    BEQ     $63
 1947
 1948  004236                                                    ;ERROR DID NOT CLEAR IN RBUF
 1949  004236  104105                    ERROR   105             ERRHRD 105
 1950
 1951  004240                                                        ENDIF
 1952  004240                        $63:
 1953  004240                                                    ENDSUB
 1954  004240                                                    EXIT
 1955  004240  000400                    BR      TST13       ;;EXIT THIS TEST
```

```
                                ;;*****************************************************************
                                ;*TEST 13       TRANSMITTER INTERRUPT LOGIC TEST
                                ;*              LOGICALLY THIS IS 4 SEPARATE TESTS
                                ;*              A) DOES TRANSMITTER INTERRUPT LOGIC WORK
                                ;*              B) AT PRIORITY OF 0
                                ;*              C) AND ONLY ONCE
                                ;*              D) BUT NOT WITH INTERRUPT ENABLE CLEAR
                                ;;*****************************************************************
004242  000004                  TST13:  SCOPE
004244  012737  000010  001160          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
004252  012737  000013  001200          MOV     #13,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
                                                                        ;CLEAR 'INTERRUPT OCCURED' FLAG
                                                                LET INTFLAG := #0
004260  005037  013750                  CLR     INTFLAG

                                                                        ;GET VECTOR ADDRESS
                                                                LET R3 := DLVEC
004264  013703  001256                  MOV     DLVEC,R3
                                                                        ;FOR THE TRANSMITTER
                                                                LET R3 := R3 + #4
004270  062703  000004                  ADD     #4,R3
                                                                        ;SET VECTOR TO POINT TO TRANS.SRV AT PRI
                                                                SETVEC R3, #INTSRV, #PR7
004274  010146                  MOV     R1,-(SP)
004276  010301                  MOV     R3,R1
004300  012721  013742          MOV     #INTSRV,(R1)+
004304  012711  000340          MOV     #PR7,(R1)
004310  012601                  MOV     (SP)+,R1
004312                                                  BGNSUB
004312  012737  004320  001110          MOV     #64$,$LPERR

                                                        ;; MAKE SURE THAT TRANSMITTER READY IS SET
004320                                                  CALL TIMER IN <#500,#XMITRDY,TCSR,#SET>
004320  010546                  MOV     R5,-(SP)
004322  012745  177777          MOV     #SET,-(R5)
004326  013745  001264          MOV     TCSR,-(R5)
004332  012745  000200          MOV     #XMITRDY,-(R5)
004336  012745  000500          MOV     #500,-(R5)
004342  004737  012560          JSR     PC,TIMER
004346  012605                  MOV     (SP)+,R5


                                                            ;CLEAR INTERRUPT ENABLE
004350                                                  LET @TCSR := @TCSR CLR.BY #XMITIE
004350  042777  000100  174706          BIC     #XMITIE,@TCSR


                                                            ;SET IT TO 0
004356                          SETPRI #PR0


                                                            ;NOW SET I.E. BIT
004370                                                  LET @TCSR := @TCSR SET.BY #XMITIE
004370  052777  000100  174666          BIS     #XMITIE,@TCSR

004376                                                  CALL TIMER IN <#500,#1,#INTFLAG,#SET>
004376  010546                  MOV     R5,-(SP)
004400  012745  177777          MOV     #SET,-(R5)
004404  012745  013750          MOV     #INTFLAG,-(R5)
```

```
2012  004410  012745  000001          MOV     #1,-(R5)
2013  004414  012745  000500          MOV     #500,-(R5)
2014  004420  004737  012560          JSR     PC,TIMER
2015  004424  012605                  MOV     (SP)+,R5
2016
2017                                                        ;DID IT SET IN TIME?
2018  004426                                              IF.ERROR THEN
2019  004426  103001                  BCC     $64
2020                                                        ;INTERRUPT DID NOT OCCUR
2021  004430                                                ERRHRD 106
2022  004430  104106                  ERROR   106
2023  004432                                              ENDIF
2024  004432                  $64:
2025
2026                                                        ;LET POSSIBLE 2'ND INTERR OCCUR
2027  004432                                              WAITMS 500.
2028  004432  010546                  MOV     R5,-(SP)
2029  004434  012745  000764          MOV     #500.,-(R5)
2030  004440  004737  012732          JSR     PC,WAIT
2031  004444  012605                  MOV     (SP)+,R5
2032
2033                                                        ;DID EXACTLY 1 INTERRUPT OCCUR
2034  004446                                              IF INTFLAG GT #1 THEN
2035  004446  023727  013750  000001  CMP     INTFLAG,#1
2036  004454  003401                  BLE     $65
2037                                                        ;TRANSMITTER INTERRUPTED TWICE
2038  004456                                                  ERRHRD 107
2039  004456  104107                  ERROR   107
2040  004460                                              ENDIF
2041  004460                  $65:
2042  004460                                            ENDSUB
2043                                          ;INTERRUPT WITHOUT INTERRUPT ENABLE SET
2044  004460                                            BGNSUB
2045  004460  012737  004466  001110  MOV     #64$,$LPERR
2046                                                        ;CLEAR INTERRUPT ENABLE
2047  004466                                              LET @TCSR := @TCSR CLR.BY #XMITIE
2048  004466  042777  000100  174570  BIC     #XMITIE,@TCSR
2049                                                        ;CLEAR 'INTERRUPT OCCURED' FLAG
2050  004474                                              LET INTFLAG := #0
2051  004474  005037  013750          CLR     INTFLAG
2052                                                  ;NO INTERRUPTS SHOULD OCCUR, PSW STILL AT 0.
2053                                                  ;DARE IT TO HAPPEN
2054  004500                                                  LET @TBUF := #0
2055  004500  005077  174562          CLR     @TBUF
2056                                                ;SEE IF INT FLAG EVER SETS
2057  004504                                        CALL TIMER IN <#1000,#1,#INTFLAG,#SET>
2058  004504  010546                  MOV     R5,-(SP)
2059  004506  012745  177777          MOV     #SET,-(R5)
2060  004512  012745  013750          MOV     #INTFLAG,-(R5)
2061  004516  012745  000001          MOV     #1,-(R5)
2062  004522  012745  001000          MOV     #1000,-(R5)
2063  004526  004737  012560          JSR     PC,TIMER
2064  004532  012605                  MOV     (SP)+,R5
2065                                                ;DID IT SET DURING TIMER?
2066  004534  103401                  BCS     1$    ;BR IF NO
2067                                                      ;INTERR STILL OCCURED WITH IE DISABLED
```

```
                                                                ERRHRD 110
004536
004536  104110                  ERROR   110
004540                  1$:
004540  000005                  RESET
004542                                                      ENDSUB
                                                                ;RESTORE VECTOR AREA
004542                                                      CLRVEC R3
004542  010146                  MOV     R1,-(SP)
004544  010246                  MOV     R2,-(SP)
004546  012701  000003          MOV     #R3,R1
004552  010102                  MOV     R1,R2
004554  062702  000002          ADD     #2,R2
004560  010221                  MOV     R2,(R1)+
004562  005011                  CLR     (R1)
004564  012602                  MOV     (SP)+,R2
004566  012601                  MOV     (SP)+,R1
```

```
2085
2086                                    ;;*****************************************************************
2087                                    ;*TEST 14       RECEIVER INTERRUPT LOGIC TEST
2088                                    ;*             THIS TEST COVERS ALL OF THE RECEIVER
2089                                    ;*             SIDE OF THE INTERRUPT LOGIC IN
2090                                    ;*             CHARACTER MODE.
2091                                    ;;*****************************************************************
2092  004570  000004                    TST14:  SCOPE
2093  004572  012737  000010  001160            MOV     #10,$TIMES      ;;DO 10 ITERATIONS
2094  004600  012737  000014  001200            MOV     #14,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2095  004606                                                                    IF #WRAP NOTSETIN $USWR OR CONSOL EQ #TRUE THEN
2096  004606  032737  000020  001220            BIT     #WRAP,$USWR
2097  004614  001404                            BEQ     $66
2098  004616  023727  013736  000001            CMP     CONSOL,#TRUE
2099  004624  001002                            BNE     $67
2100  004626                                $66:
2101  004626  000137  005156                    JMP     TST15                   ;       EXIT TEST
2102  004632                                                                    ENDIF
2103  004632                                $67:
2104
2105                                                                                ;CLEAR INTERRUPT OCCURED FLAG
2106                                                                                ;SET UP RECEIVER INTER.VECTOR
2107  004632                                                                    SETVEC DLVEC,#INTSRV,#PR7
2108  004632  010146                            MOV     R1,-(SP)
2109  004634  013701  001256                    MOV     DLVEC,R1
2110  004640  012721  013742                    MOV     #INTSRV,(R1)+
2111  004644  012711  000340                    MOV     #PR7,(R1)
2112  004650  012601                            MOV     (SP)+,R1
2113                                            ;PRIORITY 0 AND MULTIPLE INTERRUPT TEST.-RCVRIE
2114  004652                                                                    BGNSUB
2115  004652  012737  004660  001110            MOV     #64$,$LPERR
2116  004660                                                                    LET INTFLAG := #0
2117  004660  005037  013750                    CLR     INTFLAG
2118                                                                                ;CLEAR INTERRUPTS
2119  004664                                                                    LET @RCSR := @RCSR CLR.BY #RCVRIE
2120  004664  042777  000100  174366            BIC     #RCVRIE,@RCSR
2121                                                                                ;CHANGE PRIORITY
2122                                                                                ;..TO 0
2123  004672                                SETPRI #PR0
2124
2125                                                                                ;SEND A CHARACTER
2126  004704                                                                    LET @TBUF :B= #0
2127  004704  105077  174356                    CLRB    @TBUF
2128                                                                                ;WAIT A MAXIMUM
2129                                                                            ;OF 50 MSEC FOR
2130                                                                            ;XMIT RDY TO SET IN TCSR
2131  004710                                                                    CALL TIMER IN <#500,#RCVRDONE,TCSR,#SET>
2132  004710  010546                            MOV     R5,-(SP)
2133  004712  012745  177777                    MOV     #SET,-(R5)
2134  004716  013745  001264                    MOV     TCSR,-(R5)
2135  004722  012745  000200                    MOV     #RCVRDONE,-(R5)
2136  004726  012745  000500                    MOV     #500,-(R5)
2137  004732  004737  012560                    JSR     PC,TIMER
2138  004736  012605                            MOV     (SP)+,R5
2139                                                                                ;SET INTERRUPT ENABLE
2140  004740                                                                    LET @RCSR := @RCSR SET.BY #RCVRIE
```

```
 2141  004740  052777  000100  174312          BIS     #RCVRIE,@RCSR
 2142                                                          ;LET IT COME IN.
 2143
 2144  004746                                                CALL TIMER IN <#500,#1,#INTFLAG,#SET>
 2145  004746  010546                          MOV     R5,-(SP)
 2146  004750  012745  177777                  MOV     #SET,-(R5)
 2147  004754  012745  013750                  MOV     #INTFLAG,-(R5)
 2148  004760  012745  000001                  MOV     #1,-(R5)
 2149  004764  012745  000500                  MOV     #500,-(R5)
 2150  004770  004737  012560                  JSR     PC,TIMER
 2151  004774  012605                          MOV     (SP)+,R5
 2152
 2153                                                          ;DID IT SET IN TIME?
 2154  004776                                                IF.ERROR THEN
 2155  004776  103001                          BCC     $70
 2156                                                          ;INTERRUPT DID NOT OCCUR
 2157  005000                                                  ERRHRD 111
 2158  005000  104111                          ERROR   111
 2159  005002                                                ENDIF
 2160  005002                          $70:
 2161                                                      ;LET POSSIBLE 2'ND INTERR OCCUR
 2162  005002                                              WAITMS 500
 2163  005002  010546                          MOV     R5,-(SP)
 2164  005004  012745  000500                  MOV     #500,-(R5)
 2165  005010  004737  012732                  JSR     PC,WAIT
 2166  005014  012605                          MOV     (SP)+,R5
 2167
 2168                                                          ;EXACTLY 1 INTERRUPT?
 2169  005016                                                IF INTFLAG GT #1 THEN
 2170  005016  023727  013750  000001          CMP     INTFLAG,#1
 2171  005024  003401                          BLE     $71
 2172                                                              ;RECEIVER INTERRUPTED TWICE
 2173  005026                                                          ERRHRD 112
 2174  005026  104112                          ERROR   112
 2175  005030                                                    ENDIF
 2176  005030                          $71:
 2177  005030                                                ENDSUB
 2178
 2179
 2180
 2181
 2182
 2183
 2184                                                          ;INTERRUPT WITHOUT IE SET.
 2185  005030                                                  BGNSUB
 2186  005030  012737  005036  001110          MOV     #64$,$LPERR
 2187
 2188                                                              ;CLEAR INTERRUPT
 2189  005036                                                  LET @RCSR := @RCSR CLR.BY #RCVRIE
 2190  005036  042777  000100  174214          BIC     #RCVRIE,@RCSR
 2191                                                              ;CLEAR INTERRUPT FLAG
 2192  005044                                                  LET INTFLAG := #0
 2193  005044  005037  013750                  CLR     INTFLAG
 2194                                                              ; SEND A CHARACTER
 2195  005050                                                  LET @TBUF :B= #0
 2196  005050  105077  174212                  CLRB    @TBUF
```

```
                                                                ; DARE IT
                                                                
005054                                                          CALL TIMER IN <#500,#1,#INTFLAG,#CLR>
005054  010546                 MOV     R5,-(SP)
005056  012745  000000         MOV     #CLR,-(R5)
005062  012745  013750         MOV     #INTFLAG,-(R5)
005066  012745  000001         MOV     #1,-(R5)
005072  012745  000500         MOV     #500,-(R5)
005076  004737  012560         JSR     PC,TIMER
005102  012605                 MOV     (SP)+,R5
                                                                ;DID IT CLEAR IN TIME?
005104                                                          IF.ERROR THEN
005104  103001                 BCC     $72
                                                                ;INTERR STILL OCCURED WITH IE DISABLED
005106                                                          ERRHRD 113
005106  104113                 ERROR   113
005110                                                          ENDIF
005110                  $72:
005110                                                          ENDSUB
005110                  SETPRI  #PR7    ;RAISE CPU PRIORITY
                                                                ;CLEAR THE WORLD
005122                                                              BRESET
005122  000005                 RESET


005124                                                          LET R4 := @DLVEC
005124  017704  174126         MOV     @DLVEC,R4
005130                                                          CLRVEC R4
005130  010146                 MOV     R1,-(SP)
005132  010246                 MOV     R2,-(SP)
005134  012701  000004         MOV     #R4,R1
005140  010102                 MOV     R1,R2
005142  062702  000002         ADD     #2,R2
005146  010221                 MOV     R2,(R1)+
005150  005011                 CLR     (R1)
005152  012602                 MOV     (SP)+,R2
005154  012601                 MOV     (SP)+,R1
```

```
                                ;;*****************************************************************
                                ;*TEST 15       TEST DATA  WRAP AROUND: FLAG MODE
                                ;;*****************************************************************
005156  000004                  TST15:  SCOPE
005160  012737  000001  001160          MOV     #1,$TIMES       ;;DO 1 ITERATION
005166  012737  000015  001200          MOV     #15,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
005174                                          IF #WRAP NOTSETIN $USWR OR CONSOL EQ #TRUE THEN
005174  032737  000020  001220          BIT     #WRAP,$USWR
005202  001404                          BEQ     $73
005204  023727  013736  000001          CMP     CONSOL,#TRUE
005212  001001                          BNE     $74
005214                          $73:
                                                                ;CAN'T TEST WITHOUT A WRAP
005214                                              EXIT
005214  000506                          BR      TST16           ;;EXIT THIS TEST
005216                                          ENDIF
005216                          $74:
005216                                          BRESET
005216  000005                          RESET
                                                                ;BINARY COUNT PATTERN
005220                                          INCR R2 FROM #0 TO #377 BY #1
005220  005002                          CLR     R2
005222  000401                          BR      $75
005224                          $76:
005224  005202                          INC     R2
005226                          $75:
005226  020227  000377                  CMP     R2,#377
005232  003077                          BGT     $77
                                ;
                                ;
                                ;
                                                    ; MAKE SURE IT'S READY
005234                                          CALL TIMER IN <#500,#XMITRDY,TCSR,#SET>
005234  010546                          MOV     R5,-(SP)
005236  012745  177777                  MOV     #SET,-(R5)
005242  013745  001264                  MOV     TCSR,-(R5)
005246  012745  000200                  MOV     #XMITRDY,-(R5)
005252  012745  000500                  MOV     #500,-(R5)
005256  004737  012560                  JSR     PC,TIMER
005262  012605                          MOV     (SP)+,R5
005264                                          IF.ERROR THEN
005264  103002                          BCC     $100
                                                    ; TRANSMITTER NEVER BECAME READY
005266                                              ERRHRD 123
005266  104123                          ERROR   123
005270                                              EXIT
005270  000460                          BR      TST16           ;;EXIT THIS TEST
005272                                          ENDIF
005272                          $100:
                                ;
                                                    ;START IT ON ITS WAY
005272                                          LET @TBUF :B= R2
005272  110277  173770                  MOVB    R2,@TBUF
                                ;
005276                                          ; NOW WAIT FOR RECIEVER DONE
                                                CALL TIMER IN <#500,#RCVRDONE,RCSR,#SET>
```

```
2290  005276  010546                    MOV     R5,-(SP)
2291  005300  012745  177777            MOV     #SET,-(R5)
2292  005304  013745  001260            MOV     RCSR,-(R5)
2293  005310  012745  000200            MOV     #RCVRDONE,-(R5)
2294  005314  012745  000500            MOV     #500,-(R5)
2295  005320  004737  012560            JSR     PC,TIMER
2296  005324  012605                    MOV     (SP)+,R5
2297  005326                                                          IF.ERROR THEN
2298  005326  103002                    BCC     $101
2299  005330                                                              ERRHRD 124
2300  005330  104124                    ERROR   124
2301                                                                  ; RECIEVER NEVER BECAME READY
2302  005332                                                          EXIT
2303  005332  000437                    BR      TST16           ;;EXIT THIS TEST
2304  005334                                                          ENDIF
2305  005334                    $101:
2306
2307                                                                  ;RETRIEVE
2308  005334                                                      LET R3 := @RBUF
2309  005334  017703  173722            MOV     @RBUF,R3
2310                                                                  ;CHECK FOR ERROR DURING TRANSFER
2311  005340                                                      IF #ERROR SETIN R3 THEN
2312  005340  032703  100000            BIT     #ERROR,R3
2313  005344  001401                    BEQ     $102
2314                                                                   ;ERROR BIT SET IN HIGH BYTE OF RBUF
2315  005346                                                           ERRHRD 200
2316  005346  104200                    ERROR   200
2317  005350                                                       ENDIF
2318  005350                    $102:
2319
2320
2321                                                                  ;COMPARE DATA
2322  005350                                                    IF R3 NE R2 THEN
2323  005350  020302                        CMP     R3,R2
2324  005352  001415                        BEQ     $103
2325  005354                                                       IF #BIT0 NOTSETIN $USWR THEN
2326  005354  032737  000001  001220        BIT     #BIT0,$USWR
2327  005362  001006                        BNE     $104
2328  005364                                                           IF R2 LT #200 THEN
2329  005364  020227  000200                CMP     R2,#200
2330  005370  002002                        BGE     $105
2331                                                                      ;DATA COMPARE ERR IN 7 BIT WORD
2332  005372                                                            ERRHRD 117
2333  005372  104117                        ERROR   117
2334  005374                                                            EXIT
2335  005374  000416                        BR      TST16           ;;EXIT THIS TEST
2336  005376                                                           ENDIF
2337  005376                        $105:
2338  005376                                                       ELSE
2339  005376  000402                        BR      $106
2340  005400                        $104:
2341                                                                   ;DATA COMP ERR IN 8 BIT WORD
2342  005400                                                           ERRHRD 17
2343  005400  104017                        ERROR   17
2344  005402                                                           EXIT
2345  005402  000413                        BR      TST16           ;;EXIT THIS TEST
```

```
                                                                                    ENDIF
                                                                                
                                                                                ELSE IF #BIT0 NOTSETIN $USWR AND R2 GT #177 THEN
                                                                                
                                                                                
                                                                                
                                                                                
                                                                                
                                                                                
                                                                                    ;GETTING 8 BITS ON 7 BIT XMIT
                                                                                    ;MAKE SURE $USWR SETUP CORRECTLY.
                                                                                    ERRHRD 22
                                                                                
                                                                                    EXIT
                                                                                ;;EXIT THIS TEST
                                                                                    ENDIF
                                                                                
                                                                                
                                                                                ENDINC  ;R2
                                                                                
```

```
 2367
 2368                                   ;;*****************************************************************
 2369                                   ;*TEST 16       TEST DATA WRAP AROUND: INTERRUPT MODE
 2370                                   ;;*****************************************************************
 2371  005432  000004                  TST16:  SCOPE
 2372  005434  012737  000001  001160          MOV     #1,$TIMES       ;;DO 1 ITERATION
 2373  005442  012737  000016  001200          MOV     #16,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
 2374  005450                                                                  IF #WRAP SETIN $USWR OR CONSOL EQ #TRUE THEN
 2375  005450  032737  000020  001220          BIT     #WRAP,$USWR
 2376  005456  001004                          BNE     $111
 2377  005460  023727  013736  000001          CMP     CONSOL,#TRUE
 2378  005466  001002                          BNE     $112
 2379  005470                          $111:
 2380  005470  000137  006102                  JMP     TST17                           ;EXIT TEST
 2381  005474                                                                  ENDIF
 2382  005474                          $112:
 2383
 2384                                          ; THIS TEST WILL RUN BOTH TRANSMITTER AND
 2385                                          ; RECIEVER AT FULL SPEED TESTING
 2386                                          ; THE ABILITY OF THE MODULE
 2387                                          ; TO HANDLE INTERRUPTS FROM BOTH SIDES AT ONCE.
 2388                                          ;
 2389                                          ;DOUBLE BUFFERING IS NOT FULLY TESTED BECAUSE OF
 2390                                          ;APT CONSIDERATIONS. I.E. 'BREAK' FROM APT
 2391                                          ;CAUSES OVERRUN ERRORS. THEREFORE TRANSMIT INTR IS
 2392                                          ;ENABLED ONLY AFTER THE RECVR HAS OBTAINED THE LAST WORD
 2393
 2394                                          ;  THIS TEST WILL TRANSFER A MAXIMUM OF 400(8)
 2395                                          ; CHARACTERS THROUGH THE MODULE, BUT IF AN ERROR
 2396                                          ; IS DETECTED BY THE TEST A PREMATURE SHUTDOWN OCCURS.
 2397
 2398
 2399                                                                          ;CHANGE PRIORITY
 2400  005474                          SETPRI #PR0                             ;..TO 0
 2401
 2402  005506                                                                  ;GET VECTOR ADDRESS
 2403  005506  013701  001256                  MOV     DLVEC,R1        LET R1 := DLVEC
 2404
 2405  005512                                                                  ;RCVR VECTOR
 2406  005512  012721  005774                  MOV     #REC,(R1)+      LET (R1)+ := #REC
 2407  005516
 2408  005516  012721  000340                  MOV     #PR7,(R1)+      LET (R1)+ := #PR7
 2409
 2410                                                                          ;POINT TO TRANSMITTER VECTOR
 2411  005522                                                                  ;AND SET IT UP ALSO
 2412  005522  012721  005736                  MOV     #TRAN,(R1)+     LET (R1)+ := #TRAN
 2413  005526
 2414  005526  012711  000340                  MOV     #PR7,(R1)       LET (R1)  := #PR7
 2415
 2416
 2417  005532                                                                  ; CLEAR ERROR COUNTER
 2418  005532  005037  005732                  CLR     ERRCNT          LET ERRCNT := #0
 2419  005536
 2420  005536  005037  005772                  CLR     DATA            LET DATA := #0  ;XMIT DATA
 2421  005542
 2422  005542  032737  000001  001220          BIT     #BIT0,$USWR     IF #BIT0 NOTSETIN $USWR THEN
```

```
2423  005550  001004                    BNE     $113
2424  005552                                                    LET NUMBER := #200
2425  005552  012737  000200  005734    MOV     #200,NUMBER
2426  005560                                                  ELSE
2427  005560  000403                    BR      $114
2428  005562                          $113:
2429  005562                                                    LET NUMBER := #400
2430  005562  012737  000400  005734    MOV     #400,NUMBER
2431  005570                                                  ENDIF
2432  005570                          $114:
2433
2434  005570                                                  BRESET  ;SET UP ALL REGISTERS
2435  005570  000005                    RESET
2436
2437                                                                ;SET I.E. IN TRANSMITTER
2438  005572                                                  LET @TCSR := @TCSR SET.BY #XMITIE
2439  005572  052777  000100  173464    BIS     #XMITIE,@TCSR
2440                                                                ;AND RECEIVER
2441  005600                                                  LET @RCSR := @RCSR SET.BY #RCVRIE
2442  005600  052777  000100  173452    BIS     #RCVRIE,@RCSR
2443
2444
2445                                                        ;NOW WE WAIT
2446  005606                                                  REPEAT
2447  005606                          $115:
2448  005606                                                  UNTIL DATA EQ NUMBER OR ERRCNT GT #0
2449  005606  023737  005772  005734    CMP     DATA,NUMBER
2450  005614  001403                    BEQ     $116
2451  005616  005737  005732            TST     ERRCNT
2452  005622  003771                    BLE     $115
2453  005624                          $116:
2454
2455                                                        ;NOW LETS CHECK.
2456                                                        ;TURN OFF ALL INTR ENABLE
2457  005624                                                LET @TCSR := @TCSR CLR.BY #XMITIE
2458  005624  042777  000100  173432    BIC     #XMITIE,@TCSR
2459  005632                                                LET @RCSR := @RCSR CLR.BY #RCVRIE
2460  005632  042777  000100  173420    BIC     #RCVRIE,@RCSR
2461  005640                                                IF ERRCNT NE #0 THEN
2462  005640  005737  005732            TST     ERRCNT
2463  005644  001431                    BEQ     $117
2464  005646                                                   IF #ERROR SETIN RHLD THEN
2465  005646  032737  100000  006100    BIT     #ERROR,RHLD
2466  005654  001424                    BEQ     $120
2467  005656                                                      IF #ORERR SETIN RHLD THEN
2468  005656  032737  040000  006100    BIT     #ORERR,RHLD
2469  005664  001402                    BEQ     $121
2470                                                                ;OVERRUN ERROR
2471  005666                                                        ERRHRD 220
2472  005666  104220                    ERROR   220
2473  005670                                                      ELSE IF #FRERR SETIN RHLD THEN
2474  005670  000415                    BR      $122
2475  005672                          $121:
2476  005672  032737  020000  006100    BIT     #FRERR,RHLD
2477  005700  001402                    BEQ     $123
2478                                                                ;FRAMING ERROR
```

```
2479  005702
2480  005702  104221                    ERROR   221
2481  005704
2482  005704  000407                    BR      $124
2483  005706                    $123:
2484  005706  032737  010000  006100    BIT     #PERR,RHLD
2485  005714  001402                    BEQ     $125
2486
2487  005716
2488  005716  104222                    ERROR   222
2489  005720
2490  005720  000401                    BR      $126
2491  005722                    $125:
2492
2493  005722
2494  005722  104024                    ERROR   24
2495  005724
2496  005724                    $126:
2497  005724                    $124:
2498  005724                    $122:
2499  005724
2500  005724  000401                    BR      $127
2501  005726                    $120:
2502
2503  005726
2504  005726  104120                    ERROR   120
2505  005730
2506  005730                    $127:
2507  005730
2508  005730                    $117:
2509  005730
2510  005730  000464                    BR      TST17
2511
2512
2513  005732  000000          ERRCNT: 0
2514  005734  000000          NUMBER: 0
```

```
        ERRHRD 221
    ELSE IF #PERR SETIN RHLD THEN

        ;PARITY ERROR
        ERRHRD 222
    ELSE

        ;UNKNOWN ERROR
        ERRHRD 24
    ENDIF


ELSE

        ;DATA COMPARE ERROR
        ERRHRD 120
ENDIF
    ENDIF
ENDIF
    EXIT    ;SKIP OVER SUPPORT ROUTINES & STORAGE
;;EXIT THIS TEST
```

```
                                ;;*****************************************************************
                                ;TRANSMIT INTERRUPT HANDLER
                                ;;*****************************************************************
                                                        BGNSRV  TRAN
005736                          TRAN:
                                                           IF DATA NE NUMBER AND ERRCNT EQ #0 THEN
005736  023737  005772  005734          CMP     DATA,NUMBER
005744  001406                        BEQ     $130
005746  005737  005732                TST     ERRCNT
005752  001003                        BNE     $130
                                                                ;SHIP OUT WORD
                                                                LET @TBUF := DATA
005754  013777  005772  173304          MOV     DATA,@TBUF
005762                                                     ENDIF
005762                          $130:
                                                           ;STOP INTERR, NOT EXER DOUBL BUFFER
                                                           LET @TCSR := @TCSR CLR.BY #XMITIE
005762  042777  000100  173274          BIC     #XMITIE,@TCSR
005770                                                  ENDSRV
005770  000002                        RTI

005772  000000          DATA:   0

                                ;;*****************************************************************
                                ;RECEIVER INTERRUPT HANDLER
                                ;;*****************************************************************
005774                                                  BGNSRV  REC
005774                          REC:
                                                                ;GET CHAR
                                                                LET RHLD := @RBUF
005774  017737  173262  006100          MOV     @RBUF,RHLD
                                                                ;CHECK ERROR
                                                                IF #ERROR SETIN RHLD OR RHLD NE  DATA THEN
006002  032737  100000  006100          BIT     #ERROR,RHLD
006010  001004                        BNE     $131
006012  023737  006100  005772          CMP     RHLD,DATA
006020  001411                        BEQ     $132
006022                          $131:
                                                           ;STOP ALL INTERR PROC & GET OUT
                                                           LET DATA := NUMBER
006022  013737  005734  005772          MOV     NUMBER,DATA
                                                           LET @RCSR := @RCSR CLR.BY #RCVRIE
006030  042777  000100  173222          BIC     #RCVRIE,@RCSR
                                                           LET ERRCNT := ERRCNT + #1
006036  005237  005732                INC     ERRCNT
                                                        ELSE
006042  000415                        BR      $133
006044                          $132:
                                                           LET DATA := DATA + #1
006044  005237  005772                INC     DATA
                                                              IF DATA EQ NUMBER THEN
006050  023737  005772  005734          CMP     DATA,NUMBER
006056  001004                        BNE     $134
                                                                LET @RCSR := @RCSR CLR.BY #RCVRIE
006060  042777  000100  173172          BIC     #RCVRIE,@RCSR
```

```
2571  006066
2572  006066  000403                    BR      $135                                    ELSE
2573  006070                        $134:
2574                                                                                    ;ALLOW NEXT XMIT INTERR
2575  006070                                                                            LET @TCSR := @TCSR SET.BY #XMITIE
2576  006070  052777  000100  173166    BIS     #XMITIE,@TCSR
2577  006076                                                                          ENDIF
2578  006076                        $135:
2579  006076                                                                ENDIF
2580  006076                        $133:
2581  006076                                                          ENDSRV
2582  006076  000002                    RTI
2583
2584  006100  000000                RHLD:   0
2585
```

```
                        ;;*****************************************************************
                        ;*TEST 17       TEST BREAK GENERATION LOGIC
                        ;*              TRANSMIT KNOWN CHAR WITH BREAK SET.
                        ;*              AND COMPARE RECEIVED WITH 0.
                        ;*              FRAMING ERROR WILL ALSO BE CHECKED
                        ;*              IF ERROR BITS ARE ENABLED.
                        ;;*****************************************************************
006102  000004         TST17:  SCOPE
006104  012737  000010  001160          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
006112  012737  000017  001200          MOV     #17,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
006120                                                                  IF #WRAP NOTSETIN $USWR OR #BRK NOTSETIN $USWR T
006120  032737  000020  001220          BIT     #WRAP,$USWR
006126  001404                          BEQ     $136
006130  032737  000010  001220          BIT     #BRK,$USWR
006136  001003                          BNE     $137
006140                          $136:
006140  000137  006614                  JMP     TST20                           ;EXIT TEST
006144                                                                  ELSE
006144  000406                          BR      $140
006146                          $137:
006146                                                                       IF CONSOL EQ #TRUE THEN
006146  023727  013736  000001          CMP     CONSOL,#TRUE
006154  001002                          BNE     $141
006156                                                                          ;CAN'T TEST CONSOLE
006156  000137  006614                  JMP     TST20                          ;EXIT TEST
006162                                                                       ENDIF
006162                          $141:
006162                                                                  ENDIF
006162                          $140:
006162                                                          BGNSUB
006162  012737  006170  001110          MOV     #64$,$LPERR
006170                                                                  LET ERRCHK := #0        ; CLEAR ERROR WORD
006170  005037  006612                  CLR     ERRCHK
006174                                                                          ;SET BREAK BIT
006174  052777  000001  173062          BIS     #BREAK,@TCSR                    LET @TCSR := @TCSR SET.BY #BREAK
006202                                                                          ;NON-ZERO CHAR. '*'
006202  012777  000252  173056          MOV     #252,@TBUF                      LET @TBUF := #252
006210                                                                          ;WAIT FOR DONE
006210  010546                          MOV     R5,-(SP)                        CALL TIMER IN <#500,#RCVRDONE,RCSR,#SET>
006212  012745  177777                  MOV     #SET,-(R5)
006216  013745  001260                  MOV     RCSR,-(R5)
006222  012745  000200                  MOV     #RCVRDONE,-(R5)
006226  012745  000500                  MOV     #500,-(R5)
006232  004737  012560                  JSR     PC,TIMER
006236  012605                          MOV     (SP)+,R5
006240                                                                  IF.ERROR THEN
006240  103001                          BCC     $142
006242                                                                      ; RECIEVER DONE DID NOT SET
006242  104115                          ERROR   115                         ERRHRD 115
006244                                                                  ENDIF
```

```
2642  006244                          $142:
2643  006244
2644  006244  017700  173012                MOV     @RBUF,R0
2645  006250
2646  006250  105700                      TSTB    R0
2647  006252  001403                      BEQ     $143
2648
2649  006254
2650  006254  052737  000001  006612      BIS     #BIT0,ERRCHK
2651  006262
2652  006262                          $143:
2653  006262
2654  006262  032700  020000              BIT     #FRERR,R0
2655  006266  001003                      BNE     $144
2656  006270
2657  006270  052737  000002  006612      BIS     #BIT1,ERRCHK
2658  006276
2659  006276                          $144:
2660  006276
2661  006276  032737  000002  001220      BIT     #PARITY,$USWR
2662  006304  001421                      BEQ     $145
2663
2664  006306
2665  006306  032737  000004  001220      BIT     #EVENODD,$USWR
2666  006314  001007                      BNE     $146
2667
2668  006316
2669  006316  032700  010000              BIT     #PERR,R0
2670  006322  001003                      BNE     $147
2671
2672  006324
2673  006324  052737  000004  006612      BIS     #BIT2,ERRCHK
2674  006332
2675  006332                          $147:
2676  006332
2677  006332  000406                      BR      $150
2678  006334                          $146:
2679  006334
2680  006334  032700  010000              BIT     #PERR,R0
2681  006340  001403                      BEQ     $151
2682  006342
2683  006342  052737  000010  005732      BIS     #BIT3,ERRCNT
2684  006350
2685  006350                          $151:
2686  006350
2687  006350                          $150:
2688  006350
2689  006350                          $145:
2690
2691  006350
2692  006350  000005                      RESET
2693  006352  032777  170000  172702      BIT     #170000,@RBUF
2694  006360  001401                      BEQ     1$
2695  006362  104033                      ERROR   33
2696  006364                          1$:
2697
```

```
LET R0 := @RBUF

IFB R0 NE #0 THEN


        ; BREAK DID NOT EQUAL 0
    LET ERRCHK := ERRCHK SET.BY #BIT0

ENDIF

IF #FRERR NOTSETIN R0 THEN


        LET ERRCHK := ERRCHK SET.BY #BIT1

ENDIF

IF #PARITY SETIN $USWR THEN


            ;ODD PARITY ENABLED
    IF #EVENODD NOTSETIN $USWR THEN


            ;BREAK SHOULD GENERATE A PARITY ERRO
        IF #PERR NOTSETIN R0 THEN


                ;NO PAR ERROR WHEN THERE SHOULD
            LET ERRCHK := ERRCHK SET.BY #BIT2

        ENDIF

    ELSE

        IF #PERR SETIN R0 THEN


            LET ERRCNT := ERRCNT SET.BY #BIT3

        ENDIF

    ENDIF

ENDIF


BRESET  ;CLEAN UP


;RESET DID NOT CLEAR ERROR,FR ERR,OR PERR IN RBUF
```

```
2698  006364
2699  006364  032737  000001  006612          BIT     #BIT0,ERRCHK
2700  006372  001401                          BEQ     $152
2701  006374
2702  006374  104121                          ERROR   121
2703  006376
2704  006376                          $152:
2705  006376
2706  006376  032737  000002  006612          BIT     #BIT1,ERRCHK
2707  006404  001401                          BEQ     $153
2708  006406
2709  006406  104122                          ERROR   122
2710  006410
2711  006410                          $153:
2712  006410
2713  006410  032737  000004  006612          BIT     #BIT2,ERRCHK
2714  006416  001401                          BEQ     $154
2715  006420
2716  006420  104235                          ERROR   235
2717
2718
2719  006422
2720  006422                          $154:
2721  006422
2722  006422  032737  000010  006612          BIT     #BIT3,ERRCHK
2723  006430  001401                          BEQ     $155
2724  006432
2725  006432  104236                          ERROR   236
2726
2727
2728
2729  006434
2730  006434                          $155:
2731  006434                                                  ENDSUB
2732  006434                                                  BGNSUB
2733  006434  012737  006442  001110          MOV     #64$,$LPERR
2734
2735  006442
2736  006442  052777  000001  172614          BIS     #BREAK,@TCSR
2737
2738  006450
2739  006450  010546                          MOV     R5,-(SP)
2740  006452  012745  177777                  MOV     #SET,-(R5)
2741  006456  013745  001264                  MOV     TCSR,-(R5)
2742  006462  012745  000001                  MOV     #BREAK,-(R5)
2743  006466  012745  000500                  MOV     #500,-(R5)
2744  006472  004737  012560                  JSR     PC,TIMER
2745  006476  012605                          MOV     (SP)+,R5
2746
2747
2748  006500
2749  006500  103001                          BCC     $156
2750
2751  006502
2752  006502  104021                          ERROR   21
2753  006504
```

```
IF #BIT0 SETIN ERRCHK THEN

    ERRHRD 121  ;BREAK ERROR
ENDIF
IF #BIT1 SETIN ERRCHK THEN

    ERRHRD 122  ; FRAMING ERROR
ENDIF
IF #BIT2 SETIN ERRCHK THEN

    ERRHRD  235

        ;NO PARITY ERROR WHEN
        ;THERE SHOULD BE
ENDIF
IF #BIT3 SETIN ERRCHK THEN

    ERRHRD 236

        ;PARITY ERROR SHOULD NOT HAVE
        ;OCCURED WITH EVEN PARITY
        ;ENABLED AND BREAK SET
ENDIF


        ;SET BREAK BIT
LET @TCSR := @TCSR SET.BY #BREAK

CALL TIMER IN <#500,#BREAK,TCSR,#SET>




        ;DID IT SET?
IF.ERROR THEN

        ;BREAK DID NOT SET
        ERRHRD 21

ENDIF
```

```
2754  006504                          $156:
2755
2756                                                        ;CLEAR  BREAK BIT
2757  006504                                              LET @TCSR := @TCSR CLR.BY #BREAK
2758  006504  042777  000001  172552        BIC     #BREAK,@TCSR
2759  006512
2760  006512  010546                        MOV     R5,-(SP)
2761  006514  012745  000144               MOV     #100.,-(R5)
2762  006520  004737  012732               JSR     PC,WAIT
2763  006524  012605                        MOV     (SP)+,R5
2764                                                        WAITMS 100.
2765
2766                                                        ;READ RBUF TO CLEAR ERRORS & REC DONE
2767  006526                                              LET R0 := @RBUF
2768  006526  017700  172530               MOV     @RBUF,R0
2769                                                        ;SEND CHAR
2770  006532                                              LET @TBUF := #252
2771  006532  012777  000252  172526       MOV     #252,@TBUF
2772                                                        ;WAIT FOR DONE BIT
2773  006540                                              CALL  TIMER IN <#500,#RCVRDONE,RCSR,#SET>
2774  006540  010546                        MOV     R5,-(SP)
2775  006542  012745  177777               MOV     #SET,-(R5)
2776  006546  013745  001260               MOV     RCSR,-(R5)
2777  006552  012745  000200               MOV     #RCVRDONE,-(R5)
2778  006556  012745  000500               MOV     #500,-(R5)
2779  006562  004737  012560               JSR     PC,TIMER
2780  006566  012605                        MOV     (SP)+,R5
2781  006570                                              IF.ERROR THEN
2782  006570  103001                        BCC     $157
2783                                                        ;RECEIVER NEVER CAME READY
2784  006572                                              ERRHRD 230
2785  006572  104230                        ERROR   230
2786  006574                                              ENDIF
2787  006574                          $157:
2788
2789                                                        ;WAS CHAR AFTER BREAK RECEIVED
2790  006574                                              IFB @RBUF NE #252 THEN
2791  006574  127727  172462  000252        CMPB    @RBUF,#252
2792  006602  001401                        BEQ     $160
2793                                                        ;CHAR AFTER BREAK NOT RECEIVED CORRECTLY
2794  006604                                              ERRHRD 231
2795  006604  104231                        ERROR   231
2796  006606                                              ENDIF
2797  006606                          $160:
2798  006606                                      BRESET
2799  006606  000005                        RESET
2800  006610                                      ENDSUB
2801  006610                                      EXIT
2802  006610  000401                        BR      TST20   ;;EXIT THIS TEST
2803  006612  000000                  ERRCHK: .WORD 0
```

```
                        ;;*****************************************************************
                        ;*TEST 20       NOT A TEST - SEND BACK TO LOOP
                        ;;*****************************************************************
006614  000004          TST20:  SCOPE
006616  012737  000001  001160          MOV     #1,$TIMES       ;;DO 1 ITERATION
006624                                                                  IF #BIT12 SETIN @SWR THEN
006624  032777  010000  172306          BIT     #BIT12,@SWR
006632  001513                          BEQ     $161
006634                                                                      IF PHASE2 EQ #TRUE  THEN
006634  023727  014320  000001          CMP     PHASE2,#TRUE
006642  001021                          BNE     $162
006644                                  TYPTXT  <<CRLF>/        ** PHASE 2 SUMMARY **/<CRLF>>
006704                                                                       ELSE
006704  000425                          BR      $163
006706                          $162:
006706                                                                          IF P1CNT EQ #0 THEN
006706  005737  014322                  TST     P1CNT
006712  001022                          BNE     $164
006714                                  TYPTXT  <<CRLF>/        ** PHASE 1 SUMMARY **/<CRLF>>
006754                                                                          LET P1CNT := P1CNT + #1
006754  005237  014322                  INC     P1CNT
006760                                                                          ENDIF
006760                          $164:
006760                                                                      ENDIF
006760                          $163:
006760                                  TYPTXT  <*CSR: *>
006774                                  TYPOCT  DLADD
007002                                  TYPTXT  <*,VECTOR: *>
007022                                  TYPOCT  DLVEC
007030                                  TYPTXT  <*,ERRORS: *>
007050                                  TYPDEC  $ERTTL
007056  104401  001171                  TYPE    ,$CRLF
007062                                                                  ENDIF
007062                          $161:
007062  005037  001112                  CLR     $ERTTL          ; RESET FOR NEXT DEVICE/PASS
007066                                                                  IF PHASE2 EQ #TRUE THEN
007066  023727  014320  000001          CMP     PHASE2,#TRUE
007074  001004                          BNE     $165
007076                                                                    LET PHASE2 := #0
007076  005037  014320                  CLR     PHASE2
007102                                                                    LET $UNIT := $UNIT + #1
007102  005237  001206                  INC     $UNIT
007106                                                                  ENDIF
007106                          $165:
007106  000137  001732                  JMP     LOOP            ; BACK UP TO THE BEGINNING
```

```
                                ;;*****************************************************************
                                ;*TEST 21       TEST THAT CHANNELS INTR AT ASSIGNED PRIORITY
                                ;*             INTERRUPTS WILL BE ENABLED ON ALL ACTIVE CHANNELS.
                                ;*             RECEIVER AND TRANSMITTER. THEN WE'LL CHECK TO
                                ;*             SEE IF THEY INTERRUPTED IN THE ASSIGNED SEQUENCE.
                                ;;*****************************************************************
007112  000004                  TST21:  SCOPE
007114  012737  000001  001160          MOV     #1,$TIMES       ;;DO 1 ITERATION
007122  012737  000021  001200          MOV     #21,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX


                                                                        ;CLEAR OUT INTERRUPT TABLE
007130                                                          LET R0 := #INTRTABLE
007130  012700  010056                  MOV     #INTRTABLE,R0
007134                                                          REPEAT
007134                          $166:
007134
                                                                    LET (R0)+ := #0
007134  005020                          CLR     (R0)+
007136                                                          UNTIL R0 EQ #TABEND
007136  020027  010076                  CMP     R0,#TABEND
007142  001374                          BNE     $166


                                                                        ;SET PRIORITY TO 7
007144                                  SETPRI #PR7


                                                                        ;SET UP ALL INTERRUPT VECTORS
007156                                                          LET R0 := DLVEC
007156  013700  001256                  MOV     DLVEC,R0
007162                                                          LET R1 := #RCVR0SRV
007162  012701  007536                  MOV     #RCVR0SRV,R1
007166                                                          REPEAT
007166                          $167:
007166
                                                                    LET (R0)+ := R1
007166  010120                          MOV     R1,(R0)+
007170                                                              LET (R0)+ := #PR7
007170  012720  000340                  MOV     #PR7,(R0)+
007174                                                              LET R1 := R1 + #30
007174  062701  000030                  ADD     #30,R1
007200                                                          UNTIL R1 EQ #SRVEND
007200  020127  010036                  CMP     R1,#SRVEND
007204  001370                          BNE     $167


                                                                    ;ENABLE INTERRUPTS ON ALL ACTIVE LINES,
                                                                    ;ALSO, XMIT CHAR'S TO PRIME RECEIVERS

                                                                ;COPY MASK
007206                                                          LET CHMASK := MASK
007206  013737  014314  010076          MOV     MASK,CHMASK
007214                                                          LET CHCNT := #0
007214  005037  010100                  CLR     CHCNT
007220                                                          LET R0 := DLADD
007220  013700  001254                  MOV     DLADD,R0
007224                                                          REPEAT
007224                          $170:
```

```
2906  007224
2907  007224  032737  000001  010076          BIT     #BIT0,CHMASK
2908  007232  001430                          BEQ     $171
2909
2910  007234
2911  007234  052710  000100                  BIS     #BIT6,(R0)
2912  007240
2913  007240  062700  000004                  ADD     #4,R0
2914
2915  007244
2916  007244  052710  000100                  BIS     #BIT6,(R0)
2917
2918  007250
2919  007250  012760  000252  000002          MOV     #252,2(R0)
2920
2921  007256
2922  007256  162700  000004                  SUB     #4,R0
2923
2924  007262
2925  007262  010546                          MOV     R5,-(SP)
2926  007264  012745  177777                  MOV     #SET,-(R5)
2927  007270  010045                          MOV     R0,-(R5)
2928  007272  012745  000200                  MOV     #RCVRDONE,-(R5)
2929  007276  012745  000500                  MOV     #500,-(R5)
2930  007302  004737  012560                  JSR     PC,TIMER
2931  007306  012605                          MOV     (SP)+,R5
2932
2933  007310
2934  007310  103001                          BCC     $172
2935
2936  007312
2937  007312  104023                          ERROR   23
2938  007314
2939  007314                          $172:
2940  007314
2941  007314                          $171:
2942
2943  007314
2944  007314  062700  000010                  ADD     #10,R0
2945
2946  007320
2947  007320  006237  010076                  ASR     CHMASK
2948  007324
2949  007324  005237  010100                  INC     CHCNT
2950  007330
2951  007330  023727  010100  000003          CMP     CHCNT,#3
2952  007336  001007                          BNE     $173
2953  007340
2954  007340  004737  013662                  JSR     PC,TSTCON
2955  007344
2956  007344  005737  013736                  TST     CONSOLE
2957  007350  00140_                          BEQ     $174
2958  007352
2959  007352  005237  010100                  INC     CHCNT
2960  007356
2961  007356                          $174:
```

```
IF #BIT0 SETIN CHMASK THEN

    ;SET RCSR IE
    LET (R0) := (R0) SET.BY #BIT6

    LET R0 := R0 + #4

    ;SET XCSR IE
    LET (R0) := (R0) SET.BY #BIT6

    ;LOAD XBUF
    LET 2(R0) := #252

    ;GO BACK TO RCSR
    LET R0 := R0 - #4

    ;LOOK FOR RCVR DONE
    CALL TIMER IN <#500,#RCVRDONE,R0,#SET>

            ;DID IT SET IN TIME?
    IF.ERROR THEN

            ;RCVRDONE DID NOT SET IN RCSR
            ERRHRD 23

    ENDIF

ENDIF

    LET R0 := R0 + #10

;SETUP FOR NEXT CHANNEL
LET CHMASK := CHMASK SHIFT -1

LET CHCNT := CHCNT + #1

IF CHCNT EQ #3 THEN

    CALL TSTCON

    IF CONSOLE NE #0 THEN

            LET CHCNT := CHCNT + #1

    ENDIF
```

```
                                                        ENDIF
                                $173:
                                                    UNTIL CHCNT EQ #4
007356  023727  010100  000004            CMP     CHCNT,#4
007364  001317                            BNE     $170

                                                    ;ALL XMIT & REC INTERRUPTS SHOULD BE
                                                    ;IN CONTENTION.
                                                    ;RECEIVE INTERRUPTS HAVE PRIORITY
                                                    ;OVER XMIT INTERRUPTS.
                                                    ;LOW CHANNELS HAVE PRIORITY OVER HIGH CHANNELS.
                                                    ;THEREFORE, ONCE CPU PRIORITY IS LOWERED,
                                                    ;INTERRUPTS S/B IN THE ORDER SHOWN IN THE
                                                    ;CHANNEL IDENTIFIER TABLE (RCHO:)

                                                            ;SET UP POINTER FOR SERVICE ROUTINE
007366                                              LET R4 := #INTRTABLE
007366  012704  010056                    MOV     #INTRTABLE,R4

                                                            ;LET EM GO
007372                                    SETPRI #PRO

                                                            ;DISABLE ALL INTERRUPTS
007404                                              BRESET
007404  000005                            RESET

007406                                              LET TCNT := #0
007406  005037  010104                    CLR     TCNT
                                                    ;CLEAR ERR FLAG
007412                                              LET PRIERR := #0
007412  005037  010102                    CLR     PRIERR
007416                                              LET R0 := #INTRTABLE
007416  012700  010056                    MOV     #INTRTABLE,R0
                                                    ;SETUP EXPECTED VALUE
007422                                              LET R1 := #INTRTABLE+2
007422  012701  010060                    MOV     #INTRTABLE+2,R1
007426  013737  014314  010076            MOV     MASK,CHMASK     ;GET # OF CHANNELS IN TCNT

007434  006237  010076          1$:       ASR     CHMASK
007440  103002                            BCC     2$
007442  005237  010104                    INC     TCNT            ;GET TOTAL CHANNEL CT
007446  005737  010076          2$:       TST     CHMASK
007452  001370                            BNE     1$

007454  004737  013662                    JSR     PC,TSTCON       ;CORRECT CHANNEL CT FOR CONSOLE
007460  005737  013736                    TST     CONSOLE
007464  001402                            BEQ     3$
007466  005337  010104                    DEC     TCNT
007472  006337  010104          3$:       ASL     TCNT            ;APPLY X2 MULT FACTOR

007476  005337  010104          4$:       DEC     TCNT
007502  001410                            BEQ     5$
007504  023727  010104  177777            CMP     TCNT,#-1        ;ALL 1'S ?
007512  001404                            BEQ     5$              ;BR IF YES
007514  022120                            CMP     (R1)+,(R0)+     ;R1-R0
007516  003367                            BGT     4$              ;BR IF R1>R0
```

```
 3018  007520  005237  010102                 INC     PRIERR          ;ELSE ERROR
 3019
 3020  007524                          5$:
 3021  007524                                                         IF PRIERR NE #0 THEN
 3022  007524  005737  010102                 TST     PRIERR
 3023  007530  001401                         BEQ     $175
 3024  007532                                                             ERRHRD 250
 3025  007532  104250                         ERROR   250
 3026                                                                   ;CHANNELS DID NOT INTR ACCORDING TO
 3027                                                                   ;ASSIGNED PRIORITY. THE BYTE ENTRIES
 3028                                                                   ;IN INTRTABLE: SHOULD BE IN THE ORDER
 3029                                                                   ;THAT THEY APPEAR IN THE CHANNEL #
 3030                                                                   ;TABLE .(EXCLUDING CHANNELS NOT ACTIVE)
 3031  007534                                                         ENDIF
 3032  007534                          $175:
 3033
 3034  007534                                                         EXIT
 3035  007534  000565                         BR      TST22           ;;EXIT THIS TEST
```

```
3036
3037                                    ;;*****************************************************************
3038                                    ;*              START OF SERVICE ROUTINES
3039                                    ;;*****************************************************************
3040
3041  007536                                                        BGNSRV RCVROSRV
3042  007536                            RCVROSRV:
3043                                                                        ;PUT INTR IDENTIFIER IN INTRTABLE
3044  007536                                                              LET (R4)+ := RCH0
3045  007536  013724  010036                   MOV     RCH0,(R4)+
3046                                                                        ;GENERATE CSR ADDRESS
3047  007542                                                              LET TEMP := RCSR
3048  007542  013737  001260  010106           MOV     RCSR,TEMP
3049  007550                                                              LET TEMP := TEMP + #0
3050  007550  062737  000000  010106           ADD     #0,TEMP
3051                                                                        ;ONE INTR IS ALL WE WANT FROM HERE
3052  007556                                                              LET @TEMP := @TEMP CLR.BY #BIT6
3053  007556  042777  000100  000272           BIC     #BIT6,@TEMP
3054  007564                                                        ENDSRV
3055  007564  000002                           RTI
3056
3057
3058  007566                                                        BGNSRV  XMITOSRV
3059  007566                            XMITOSRV:
3060                                                                        ;PUT INTR IDENTIFIER IN INTRTABLE
3061  007566                                                              LET (R4)+ := TCH0
3062  007566  013724  010046                   MOV     TCH0,(R4)+
3063                                                                        ;GENERATE CSR ADDRESS
3064  007572                                                              LET TEMP := RCSR
3065  007572  013737  001260  010106           MOV     RCSR,TEMP
3066  007600                                                              LET TEMP := TEMP + #4
3067  007600  062737  000004  010106           ADD     #4,TEMP
3068                                                                        ;ONE INTR IS ALL WE WANT FROM HERE
3069  007606                                                              LET @TEMP := @TEMP CLR.BY #BIT6
3070  007606  042777  000100  000272           BIC     #BIT6,@TEMP
3071  007614                                                        ENDSRV
3072  007614  000002                           RTI
3073
3074  007616                                                        BGNSRV  RCVR1SRV
3075  007616                            RCVR1SRV:
3076                                                                        ;PUT INTR IDENTIFIER IN INTRTABLE
3077  007616                                                              LET (R4)+ := RCH1
3078  007616  013724  010040                   MOV     RCH1,(R4)+
3079                                                                        ;GENERATE CSR ADDRESS
3080  007622                                                              LET TEMP := RCSR
3081  007622  013737  001260  010106           MOV     RCSR,TEMP
3082  007630                                                              LET TEMP := TEMP + #10
3083  007630  062737  000010  010106           ADD     #10,TEMP
3084                                                                        ;ONE INTR IS ALL WE WANT FROM HERE
3085  007636                                                              LET @TEMP := @TEMP CLR.BY #BIT6
3086  007636  042777  000100  000242           BIC     #BIT6,@TEMP
3087  007644                                                        ENDSRV
3088  007644  000002                           RTI
3089
3090  007646                                                        BGNSRV  XMIT1SRV
3091  007646                            XMIT1SRV:
```

```
3092
3093  007646                                                    ;PUT INTR IDENTIFIER IN INTRTABLE
3094  007646  013724  010050          MOV     TCH1,(R4)+      LET (R4)+ := TCH1
3095                                                            ;GENERATE CSR ADDRESS
3096  007652                                                    LET TEMP := RCSR
3097  007652  013737  001260  010106  MOV     RCSR,TEMP
3098  007660                                                    LET TEMP := TEMP + #14
3099  007660  062737  000014  010106  ADD     #14,TEMP
3100                                                            ;ONE INTR IS ALL WE WANT FROM HERE
3101  007666                                                    LET @TEMP := @TEMP CLR.BY #BIT6
3102  007666  042777  000100  000212  BIC     #BIT6,@TEMP
3103  007674                                          ENDSRV
3104  007674  000002                  RTI
3105
3106  007676                                          BGNSRV  RCVR2SRV
3107  007676                  RCVR2SRV:
3108                                                            ;PUT INTR IDENTIFIER IN INTRTABLE
3109  007676                                                    LET (R4)+ := RCH2
3110  007676  013724  010042          MOV     RCH2,(R4)+
3111                                                            ;GENERATE CSR ADDRESS
3112  007702                                                    LET TEMP := RCSR
3113  007702  013737  001260  010106  MOV     RCSR,TEMP
3114  007710                                                    LET TEMP := TEMP + #20
3115  007710  062737  000020  010106  ADD     #20,TEMP
3116                                                            ;ONE INTR IS ALL WE WANT FROM HERE
3117  007716                                                    LET @TEMP := @TEMP CLR.BY #BIT6
3118  007716  042777  000100  000162  BIC     #BIT6,@TEMP
3119  007724                                          ENDSRV
3120  007724  000002                  RTI
3121
3122  007726                                          BGNSRV  XMIT2SRV
3123  007726                  XMIT2SRV:
3124                                                            ;PUT INTR IDENTIFIER IN INTRTABLE
3125  007726                                                    LET (R4)+ := TCH2
3126  007726  013724  010052          MOV     TCH2,(R4)+
3127                                                            ;GENERATE CSR ADDRESS
3128  007732                                                    LET TEMP := RCSR
3129  007732  013737  001260  010106  MOV     RCSR,TEMP
3130  007740                                                    LET TEMP := TEMP + #24
3131  007740  062737  000024  010106  ADD     #24,TEMP
3132                                                            ;ONE INTR IS ALL WE WANT FROM HERE
3133  007746                                                    LET @TEMP := @TEMP CLR.BY #BIT6
3134  007746  042777  000100  000132  BIC     #BIT6,@TEMP
3135  007754                                          ENDSRV
3136  007754  000002                  RTI
3137
3138  007756                                          BGNSRV RCVR3SRV
3139  007756                  RCVR3SRV:
3140                                                            ;PUT INTR IDENTIFIER IN INTRTABLE
3141  007756                                                    LET (R4)+ := RCH3
3142  007756  013724  010044          MOV     RCH3,(R4)+
3143                                                            ;GENERATE CSR ADDRESS
3144  007762                                                    LET TEMP := RCSR
3145  007762  013737  001260  010106  MOV     RCSR,TEMP
3146  007770                                                    LET TEMP := TEMP + #30
3147  007770  062737  000030  010106  ADD     #30,TEMP
```

```
                                                                        ;ONE INTR IS ALL WE WANT FROM HERE
                                                                    LET @TEMP := @TEMP CLR.BY #BIT6
007776 042777 000100 000102          BIC     #BIT6,@TEMP
                                                            ENDSRV
010004 000002                        RTI
                                                            BGNSRV  XMIT3SRV
                                XMIT3SRV:
                                                                        ;PUT INTR IDENTIFIER IN INTRTABLE
                                                                    LET (R4)+ := TCH3
010006 013724 010054                 MOV     TCH3,(R4)+
                                                                        ;GENERATE CSR ADDRESS
                                                                    LET TEMP := RCSR
010012 013737 001260 010106          MOV     RCSR,TEMP
                                                                    LET TEMP := TEMP + #34
010020 062737 000034 010106          ADD     #34,TEMP
                                                                        ;ONE INTR IS ALL WE WANT FROM HERE
                                                                    LET @TEMP := @TEMP CLR.BY #BIT6
010026 042777 000100 000052          BIC     #BIT6,@TEMP
                                                            ENDSRV
010034 000002                        RTI
                                SRVEND:
```

```
                                    ;**CHANNEL IDENTIFIER TABLE**
                                    ;THE ENTRIES FROM THIS
                                    ;TABLE ARE PLACED IN THE INTRTABLE
                                    ;IN THE ORDER THAT THE INTR'S
                                    ;OCCUR. (EXCLUDING NON-ACTIVE CH'S)
010036  000000              RCH0:   .WORD 0
010040  000001              RCH1:   .WORD 1
010042  000002              RCH2:   .WORD 2
010044  000003              RCH3:   .WORD 3
010046  000010              TCH0:   .WORD 10
010050  000011              TCH1:   .WORD 11
010052  000012              TCH2:   .WORD 12
010054  000013              TCH3:   .WORD 13

                                    ;THIS TABLE WILL CONTAIN ENTRIES
                                    ;REPRESENTING EACH INTR IN THE ORDER
                                    ;THAT IT OCCURED
                                    ; S/B IN THE ABOVE ORDER
                                    ;
010056  000010              INTRTABLE:      .BLKW   8.
010076                      TABEND:

010076  000000              CHMASK: .WORD 0         ;BITS 3-0 INDICATE WHICH CH'S ARE ACTIVE FOR THIS TEST
010100  000000              CHCNT:  .WORD 0         ;CHANNEL COUNTER
010102  000000              PRIERR: .WORD 0         ;ERROR WORD, CONTAINS NO. OF INTR OUT OF ORDER
010104  000000              TCNT:   .WORD 0         ;TABLE PASS CTR
010106  000000              TEMP:   .WORD 0         ;FOR SERVICE ROUTINES
```

```
3201
3202                            ;;*****************************************************************
3203                            ;*TEST 22       TEST DATA TRANSFERS WITH ALL ACTIVE LINES INTERRUPTING
3204                            ;*             IN THIS WE'LL ENABLE INTERRUPTS ON ALL CHANNELS
3205                            ;*             FOR THIS DLV11-J. THEN WE'LL XMIT AN INCREMENTING
3206                            ;*             DATA PATERN VIA INTERRUPTS AND RECORD THE RECEIVER
3207                            ;*             INTR. IN THE RECEIVER STATUS TABLE.
3208                            ;*       NOTE: DOUBLE BUFFERING CANNOT BE TESTED AT ITS MAX SPEED
3209                            ;*             BECAUSE OF APT CONSIDERATIONS. I.E. APT SENDS
3210                            ;*             'BREAKS' WHICH CAUSE OVERRUN ERRORS. THEREFORE
3211                            ;*             THE XMIT IE IS NOT ENABLED AGAIN UNTIL THE RECVR
3212                            ;*             HAS OBTAINED THE PREVIOUS WORD.
3213                            ;;*****************************************************************
3214  010110  000004            TST22:  SCOPE
3215  010112  012737  000001  001160    MOV     #1,$TIMES       ;;DO 1 ITERATION
3216  010120  012737  000022  001200    MOV     #22,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
3217                                                                ;CLR ERROR WORD,
3218  010126                                                    LET ERWRD := #0
3219  010126  005037  012500            CLR     ERWRD
3220
3221
3222                                                                ;CLEAR OUT RECEIVER TABLES
3223  010132                                                    LET R0 := #CHOTAB
3224  010132  012700  011472            MOV     #CHOTAB,R0
3225  010136                                                    REPEAT
3226  010136                            $176:
3227  010136
3228  010136  005020                    CLR     (R0)+               LET (R0)+ := #0
3229  010140
3230  010140  020027  012472            CMP     R0,#STATEND     UNTIL R0 EQ #STATEND
3231  010144  001374                    BNE     $176
3232
3233
3234  010146                                                        ;SET UP ALL VECTORS
3235  010146  012701  010772            MOV     #R0SRV,R1       LET R1 := #R0SRV
3236  010152
3237  010152  013700  001256            MOV     DLVEC,R0        LET R0 := DLVEC
3238  010156
3239  010156                            $177:                   REPEAT
3240  010156
3241  010156  010120                    MOV     R1,(R0)+            LET (R0)+ := R1
3242  010160
3243  010160  012720  000340            MOV     #PR7,(R0)+          LET (R0)+ := #PR7
3244  010164
3245  010164  062701  000054            ADD     #54,R1              LET R1 := R1 + #54
3246  010170
3247  010170  010120                    MOV     R1,(R0)+            LET (R0)+ := R1
3248  010172
3249  010172  012720  000340            MOV     #PR7,(R0)+          LET (R0)+ := #PR7
3250  010176
3251  010176  062701  000044            ADD     #44,R1              LET R1 := R1 + #44
3252  010202
3253  010202  020127  011472            CMP     R1,#SEREND      UNTIL R1 EQ #SEREND
3254  010206  001363                    BNE     $177
3255
3256  010210                            SETPRI  #PR7                ;SET PRIORTY TO 7
```

```
3257
3258
3259                                                        ;ENABLE INTR'S ON ALL ACTIVE LINES
3260  010222                                                ;THIS MODULE
3261  010222  013700  001254          MOV     DLADD,R0   LET R0 := DLADD
3262                                                            ;INITIALIZE
3263  010226                                                LET CHMSK := MASK
3264  010226  013737  014314  012472  MOV     MASK,CHMSK
3265  010234                                                LET CHCTR := #0
3266  010234  005037  012474          CLR     CHCTR
3267                                                        ,ACTIVE CHANNEL CTR
3268  010240                                                LET ACTCH := #0
3269  010240  005037  012476          CLR     ACTCH
3270  010244                                                REPEAT
3271  010244                  $200:
3272  010244                                                    IF #BIT0 SETIN CHMSK THEN
3273  010244  032737  000001  012472  BIT     #BIT0,CHMSK
3274  010252  001413          BEQ     $201
3275                                                            ;# OF ACTIVE CH
3276  010254                                                    LET ACTCH := ACTCH + #1
3277  010254  005237  012476          INC     ACTCH
3278                                                            ;SET RCSR IE
3279  010260                                                    LET (R0) := (R0) SET.BY #BIT6
3280  010260  052710  000100          BIS     #BIT6,(R0)
3281  010264                                                    LET R0 := R0 + #4
3282  010264  062700  000004          ADD     #4,R0
3283                                                            ;SET XCSR IE
3284  010270                                                    LET (R0) := (R0) SET.BY #BIT6
3285  010270  052710  000100          BIS     #BIT6,(R0)
3286  010274                                                    LET R0 := R0 + #4
3287  010274  062700  000004          ADD     #4,R0
3288  010300                                                ELSE
3289  010300  000402          BR      $202
3290  010302                  $201:                             LET R0 := R0 + #10
3291  010302
3292  010302  062700  000010          ADD     #10,R0        ENDIF
3293  010306
3294  010306                  $202:                         LET CHMSK := CHMSK SHIFT -1
3295  010306
3296  010306  006237  012472          ASR     CHMSK
3297
3298                                                        LET CHCTR := CHCTR + #1
3299  010312
3300  010312  005237  012474          INC     CHCTR        IF CHCTR EQ #3 THEN
3301  010316
3302  010316  023727  012474  000003  CMP     CHCTR,#3
3303  010324  001007          BNE     $203                     CALL TSTCON
3304  010326
3305  010326  004737  013662          JSR     PC,TSTCON        IF CONSOLE NE #0 THEN
3306  010332
3307  010332  005737  013736          TST     CONSOLE
3308  010336  001402          BEQ     $204                       LET CHCTR := CHCTR + #1
3309  010340
3310  010340  005237  012474          INC     CHCTR            ENDIF
3311  010344
3312  010344                  $204:
```

```
3313  010344                                                        ENDIF
3314  010344                        $203:
3315  010344                                                    UNTIL CHCTR EQ #4
3316  010344  023727  012474  000004          CMP     CHCTR,#4
3317  010352  001334                          BNE     $200
3318
3319                                                        ;INIT BUFFER POINTERS FOR SERVICE ROUTINES
3320  010354                                                LET R0 := #CH0TAB
3321  010354  012700  011472                  MOV     #CH0TAB,R0
3322  010360                                                LET R1 := #CH1TAB
3323  010360  012701  011672                  MOV     #CH1TAB,R1
3324  010364                                                LET R2 := #CH2TAB
3325  010364  012702  012072                  MOV     #CH2TAB,R2
3326  010370                                                LET R3 := #CH3TAB
3327  010370  012703  012272                  MOV     #CH3TAB,R3
3328
3329                                                        ;FILL XBUF WORDS
3330                                                        ;FROM TMP0 THRU TMP3E
3331                                                        ;START WORD CH 0 ALWAYS 0
3332  010374                                                LET TMP0 := #0
3333  010374  005037  012504                  CLR     TMP0
3334                                                        ;OFFSET TO INDEX
3335  010400                                                LET R4 := #0
3336  010400  005004                          CLR     R4
3337                                                        ;PTR TO TABLE TO BE FILLED
3338  010402                                                LET R5 := #TMP0E
3339  010402  012705  012506                  MOV     #TMP0E,R5
3340  010406                                                REPEAT
3341  010406                        $205:
3342  010406                                                  IF #BIT0 NOTSETIN $USWR THEN
3343  010406  032737  000001  001220          BIT     #BIT0,$USWR
3344  010414  001005                          BNE     $206
3345  010416                                                    LET (R5)+ := TABL7(R4)
3346  010416  016425  012540                  MOV     TABL7(R4),(R5)+
3347  010422                                                    LET (R5)+ := TABL7(R4)
3348  010422  016425  012540                  MOV     TABL7(R4),(R5)+
3349  010426                                                  ELSE
3350  010426  000404                          BR      $207
3351  010430                        $206:
3352  010430                                                    LET (R5)+ := TABL8(R4)
3353  010430  016425  012550                  MOV     TABL8(R4),(R5)+
3354  010434                                                    LET (R5)+ := TABL8(R4)
3355  010434  016425  012550                  MOV     TABL8(R4),(R5)+
3356  010440                                                  ENDIF
3357  010440                        $207:
3358  010440                                                    LET R4 :- R4 + #2
3359  010440  062704  000002                  ADD     #2,R4
3360  010444                                                UNTIL R4 EQ #10
3361  010444  020427  000010                  CMP     R4,#10
3362  010450  001356                          BNE     $205
3363
3364                                                        ;INIT TMP4
3365  010452                                                ;WILL BE DECR BY XMIT SERVICE ROUTINES TILL 0
3366  010452  013737  012476  012526          MOV     ACTCH,TMP4  LET TMP4 := ACTCH
3367
3368  010460                                  SETPRI #PR0           ;SET PRIORITY TO 0
```

```
 3369
 3370  010472                                                    REPEAT
 3371  010472                          $210:
 3372  010472
 3373  010472  005737  012526                TST     TMP4        UNTIL TMP4 EQ #0
 3374  010476  001375                        BNE     $210
 3375
 3376                                                            ;WAIT FOR RECV'S TO CATCH UP
 3377  010500                                                    LET TMP4 := #100000
 3378  010500  012737  100000  012526        MOV     #100000,TMP4
 3379  010506                                                    REPEAT
 3380  010506                          $211:
 3381  010506                                                       LET TMP4 := TMP4 - #1
 3382  010506  005337  012526                DEC     TMP4
 3383  010512                                                    UNTIL TMP4 EQ #0
 3384  010512  005737  012526                TST     TMP4
 3385  010516  001373                        BNE     $211
 3386
 3387                                                               ;DISABLE ALL INTR'S
 3388  010520                                                    BRESET
 3389  010520  000005                        RESET
 3390
 3391                                                               ;CHECK RECEIVER TABLE
 3392                                                               ;RESET TMP0,1,2,3
 3393  010522                                                       LET TMP0 := #0
 3394  010522  005037  012504                CLR     TMP0
 3395  010526                                                       LET TMP1 := TMP0E
 3396  010526  013737  012506  012510        MOV     TMP0E,TMP1
 3397  010534                                                       LET TMP2 := TMP1E
 3398  010534  013737  012512  012514        MOV     TMP1E,TMP2
 3399  010542                                                       LET TMP3 := TMP2E
 3400  010542  013737  012516  012520        MOV     TMP2E,TMP3
 3401
 3402  010550                                                       ;INITIALIZE
 3403  010550  005037  012474                CLR     CHCTR       LET CHCTR := #0
 3404  010554
 3405  010554  013737  014314  012472        MOV     MASK,CHMSK  LET CHMSK := MASK
 3406  010562                                                    REPEAT
 3407  010562                          $212:
 3408  010562                                                      IF #BIT0 SETIN CHMSK THEN
 3409  010562  032737  000001  012472        BIT     #BIT0,CHMSK
 3410  010570  001436                        BEQ     $213
 3411  010572                                                          !ET R0 := CHCTR
 3412  010572  013700  012474                MOV     CHCTR,R0
 3413                                                                  ;OFFSET TO INDEX (2X)
 3414  010576                                                          LET R0 :- R0 SHIFT +1
 3415  010576  006300                        ASL     R0
 3416                                                                  ;GET TABLE LOC FOR THAT CH.
 3417  010600                                                          LEi R2 := TABL6(R0)
 3418  010600  016002  012530                MOV     TABL6(R0),R2
 3419                                                                  ;OFFSET TO INDEX (4X)
 3420  010604                                                          LET R0 := R0 SHIFT +1
 3421  010604  006300                        ASL     R0
 3422                                                                  ;GET EXPECTED STARTING WORD
 3423  010606                                                          LET R1 := TMP0(R0)
 3424  010606  016001  012504                MOV     TMP0(R0),R1
```

```
3425  010612
3426  010612  062700  000002              ADD     #2,R0
3427
3428  010616
3429  010616  016003  012504              MOV     TMP0(R0),R3
3430
3431  010622
3432  010622                        $214:
3433  010622
3434  010622  032712  100000              BIT     #BIT15,(R2)
3435  010626  001403                      BEQ     $215
3436  010630
3437  010630  052737  000400  012500      BIS     #BIT8,ERWRD
3438  010636
3439  010636                        $215:
3440
3441  010636
3442  010636  121201                      CMPB    (R2),R1
3443  010640  001402                      BEQ     $216
3444  010642
3445  010642  005237  012500              INC     ERWRD
3446  010646
3447  010646                        $216:
3448
3449  010646
3450  010646  005201                      INC     R1
3451
3452  010650
3453  010650  062702  000002              ADD     #2,R2
3454
3455  010654
3456  010654  020103                      CMP     R1,R3
3457  010656  001403                      BEQ     $217
3458  010660  005737  012500              TST     ERWRD
3459  010664  001756                      BEQ     $214
3460  010666                        $217:
3461  010666
3462  010666                        $213:
3463  010666
3464  010666  005237  012474              INC     CHCTR
3465  010672
3466  010672  006237  012472              ASR     CHMSK
3467  010676
3468  010676  023727  012474  000003      CMP     CHCTR,#3
3469  010704  001010                      BNE     $220
3470  010706
3471  010706  004737  013662              JSR     PC,TSTCON
3472  010712
3473  010712  023727  013736  000001      CMP     CONSOLE,#TRUE
3474  010720  001002                      BNE     $221
3475
3476  010722
3477  010722  005237  012474              INC     CHCTR
3478  010726
3479  010726                        $221:
3480  010726
```

```
LET R0 := R0 + #2

;GET MAX WORD FOR THAT CH.
LET R3 := TMP0(R0)

REPEAT

    IF #BIT15 SETIN (R2) THEN

        LET ERWRD := ERWRD SET.BY #BIT8

    ENDIF

        ;CHECK FOR DATA COMPARE ERROR
    IFB (R2) NE R1 THEN

        LET ERWRD := ERWRD + #1

    ENDIF

        ;BUMP EXPECTED DATA WORD
        LET R1 := R1 + #1

        ;BUMP TABLE PTR
        LET R2 := R2 + #2


    UNTIL R1 EQ R3 OR ERWRD NE #0


ENDIF

 LET CHCTR := CHCTR + #1

 LET CHMSK := CHMSK SHIFT -1

    IF CHCTR EQ #3 THEN

        CALL TSTCON

       IF CONSOLE EQ #TRUE THEN

            ;EXIT
            LET CHCTR := CHCTR + #1

       ENDIF

    ENDIF
```

```
 3481  010726                          $220:
 3482  010726                                                UNTIL CHCTR EQ #4 OR ERWRD NE #0
 3483  010726  023727  012474  000004          CMP     CHCTR,#4
 3484  010734  001403                          BEQ     $222
 3485  010736  005737  012500                  TST     ERWRD
 3486  010742  001707                          BEQ     $212
 3487  010744                          $222:
 3488
 3489  010744                                                    IF #BIT8 SETIN ERWRD THEN
 3490  010744  032737  000400  012500          BIT     #BIT8,ERWRD
 3491  010752  001401                          BEQ     $223
 3492                                                          ;ERROR FLAG UP AFTER TRANSFER
 3493  010754                                                    ERRHRD  270
 3494  010754  104270                          ERROR   270
 3495  010756                                                    ENDIF
 3496  010756                          $223:
 3497  010756                                                    IFB ERWRD NE #0 THEN
 3498  010756  105737  012500                  TSTB    ERWRD
 3499  010762  001401                          BEQ     $224
 3500                                                          ;DATA COMPARE ERROR
 3501  010764                                                    ERRHRD  271
 3502  010764  104271                          ERROR   271
 3503  010766                                                    ENDIF
 3504  010766                          $224:
 3505
 3506
 3507  010766  000137  006614                  JMP     TST20                   ;EXIT TEST
```

```
3508
3509                            ;;*****************************************************************
3510                            ;*              START OF SERVICE ROUTINES
3511                            ;;*****************************************************************
3512
3513  010772                                                    BGNSRV ROSRV
3514  010772                    ROSRV:
3515  010772                                                        LET TMP := RCSR
3516  010772  013737  001260  012502        MOV     RCSR,TMP
3517  011000                                                        LET TMP := TMP + #2
3518  011000  062737  000002  012502        ADD     #2,TMP
3519  011006                                                        LET (R0)+ := @TMP
3520  011006  017720  001470                MOV     @TMP,(R0)+
3521  011012                                                        IF TMP0 NE TMP0E THEN
3522  011012  023737  012504  012506        CMP     TMP0,TMP0E
3523  011020  001407                        BEQ     $225
3524                                                                ;GO TO XCSR
3525  011022                                                        LET TMP := TMP + #2
3526  011022  062737  000002  012502        ADD     #2,TMP
3527                                                                ;ENABLE XMIT INTERR
3528  011030                                                        LET @TMP := @TMP SET.BY #BIT6
3529  011030  052777  000100  001444        BIS     #BIT6,@TMP
3530  011036                                                    ELSE
3531  011036  000402                        BR      $226
3532  011040                    $225:
3533                                                                ;ALL DONE
3534  011040                                                        LET TMP4 := TMP4 - #1
3535  011040  005337  012526                DEC     TMP4
3536  011044                                                    ENDIF
3537  011044                    $226:
3538  011044                                                ENDSRV
3539  011044  000002                        RTI
3540
3541  011046                                                    BGNSRV XOSRV
3542  011046                    XOSRV:
3543  011046                                                        LET TMP := RCSR
3544  011046  013737  001260  012502        MOV     RCSR,TMP
3545  011054                                                        LET TMP := TMP + #6
3546  011054  062737  000006  012502        ADD     #6,TMP
3547  011062                                                        LET @TMP := TMP0
3548  011062  013777  012504  001412        MOV     TMP0,@TMP
3549  011070                                                        LET TMP0 := TMP0 + #1
3550  011070  005237  012504                INC     TMP0
3551                                                                ;GO BACK TO XCSR
3552  011074                                                        LET TMP := TMP - #2
3553  011074  162737  000002  012502        SUB     #2,TMP
3554                                                                ;DISABLE XMIT INTERRUPTS,
3555                                                                ;NOT EXER DOUBLE BUFFERING
3556  011102                                                        LET @TMP := @TMP CLR.BY #BIT6
3557  011102  042777  000100  001372        BIC     #BIT6,@TMP
3558  011110                                                ENDSRV
3559  011110  000002                        RTI
3560
3561  011112                                                    BGNSRV R1SRV
3562  011112                    R1SRV:
3563  011112                                                        LET TMP := RCSR
```

```
3564  011112  013737  001260  012502          MOV     RCSR,TMP
3565  011120                                                      LET TMP := TMP + #12
3566  011120  062737  000012  012502          ADD     #12,TMP
3567  011126                                                      LET (R1)+ := @TMP
3568  011126  017721  001350                  MOV     @TMP,(R1)+
3569  011132                                                      IF TMP1 NE TMP1E THEN
3570  011132  023737  012510  012512          CMP     TMP1,TMP1E
3571  011140  001407                          BEQ     $227
3572                                                                ;GO TO XCSR
3573  011142                                                        LET TMP := TMP + #2
3574  011142  062737  000002  012502          ADD     #2,TMP
3575                                                                ;ENABLE XMIT INTERR
3576  011150                                                        LET @TMP :- @TMP SET.BY #BIT6
3577  011150  052777  000100  001324          BIS     #BIT6,@TMP
3578  011156                                                      ELSE
3579  011156  000402                          BR      $230
3580  011160                          $227:
3581                                                                ;ALL DONE
3582  011160                                                        LET TMP4 := TMP4 - #1
3583  011160  005337  012526                  DEC     TMP4
3584  011164                                                      ENDIF
3585  011164                          $230:
3586  011164                                                  ENDSRV
3587  011164  000002                          RTI
3588
3589  011166                                                  BGNSRV X1SRV
3590  011166                          X1SRV:
3591  011166                                                      LET TMP := RCSR
3592  011166  013737  001260  012502          MOV     RCSR,TMP
3593  011174                                                      LET TMP := TMP + #16
3594  011174  062737  000016  012502          ADD     #16,TMP
3595  011202                                                      LET @TMP := TMP1
3596  011202  013777  012510  001272          MOV     TMP1,@TMP
3597  011210                                                      LET TMP1 := TMP1 + #1
3598  011210  005237  012510                  INC     TMP1
3599                                                                ;GO BACK TO XCSR
3600  011214                                                      LET TMP := TMP - #2
3601  011214  162737  000002  012502          SUB     #2,TMP
3602                                                                ;DISABLE XMIT INTERRUPTS,
3603                                                                ;NOT EXER DOUBLE BUFF
3604  011222                                                      LET @TMP := @TMP CLR.BY #BIT6
3605  011222  042777  000100  001252          BIC     #BIT6,@TMP
3606  011230                                                  ENDSRV
3607  011230  000002                          RTI
3608
3609  011232                                                  BGNSRV R2SRV
3610  011232                          R2SRV:
3611  011232                                                      LET TMP := RCSR
3612  011232  013737  001260  012502          MOV     RCSR,TMP
3613  011240                                                      LET TMP := TMP + #22
3614  011240  062737  000022  012502          ADD     #22,TMP
3615  011246                                                      LET (R2)+ := @TMP
3616  011246  017722  001230                  MOV     @TMP,(R2)+
3617  011252                                                      IF TMP2 NE TMP2E THEN
3618  011252  023737  012514  012516          CMP     TMP2,TMP2E
3619  011260  001407                          BEQ     $231
```

```
                                                                    ;GO TO XCSR
                                                                    LET TMP := TMP + #2

                                                                    ;ENABLE XMIT INTERR
                                                                    LET @TMP := @TMP SET.BY #BIT6
                                                                ELSE

                                                                    ;ALL DONE
                                                                    LET TMP4 := TMP4 - #1
                                                                ENDIF
                                                            ENDSRV

                                                            BGNSRV X2SRV

                                                                LET TMP := RCSR

                                                                LET TMP := TMP + #26

                                                                LET @TMP := TMP2

                                                                LET TMP2 := TMP2 + #1

                                                                ;GO BACK TO XCSR
                                                                LET TMP := TMP - #2

                                                                ;DISABLE XMIT INTERRUPTS,
                                                                ;NOT EXER DOUBLE BUFF.
                                                                LET @TMP := @TMP CLR.BY #BIT6
                                                            ENDSRV

                                                            BGNSRV R3SRV

                                                                LET TMP := RCSR

                                                                LET TMP := TMP + #32

                                                                LET (R3)+ := @TMP

                                                                IF TMP3 NE TMP3E THEN

                                                                    ;GO TO XCSR
                                                                    LET TMP := TMP + #2

                                                                    ;ENABLE XMIT INTERR
                                                                    LET @TMP := @TMP SET.BY #BIT6
                                                                ELSE
```

```
3620
3621  011262
3622  011262  062737  000002  012502          ADD     #2,TMP
3623
3624  011270
3625  011270  052777  000100  001204          BIS     #BIT6,@TMP
3626  011276
3627  011276  000402                          BR      $232
3628  011300                          $231:
3629
3630  011300
3631  011300  005337  012526                  DEC     TMP4
3632  011304
3633  011304                          $232:
3634  011304
3635  011304  000002                          RTI
3636
3637  011306
3638  011306                          X2SRV:
3639  011306
3640  011306  013737  001260  012502          MOV     RCSR,TMP
3641  011314
3642  011314  062737  000026  012502          ADD     #26,TMP
3643  011322
3644  011322  013777  012514  001152          MOV     TMP2,@TMP
3645  011330
3646  011330  005237  012514                  INC     TMP2
3647
3648  011334
3649  011334  162737  000002  012502          SUB     #2,TMP
3650
3651
3652  011342
3653  011342  042777  000100  001132          BIC     #BIT6,@TMP
3654  011350
3655  011350  000002                          RTI
3656
3657  011352
3658  011352                          R3SRV:
3659  011352
3660  011352  013737  001260  012502          MOV     RCSR,TMP
3661  011360
3662  011360  062737  000032  012502          ADD     #32,TMP
3663  011366
3664  011366  017723  001110                  MOV     @TMP,(R3)+
3665  011372
3666  011372  023737  012520  012522          CMP     TMP3,TMP3E
3667  011400  001407                          BEQ     $233
3668
3669  011402
3670  011402  062737  000002  012502          ADD     #2,TMP
3671
3672  011410
3673  011410  052777  000100  001064          BIS     #BIT6,@TMP
3674  011416
3675  011416  000402                          BR      $234
```

```
3676  011420                        $233:
3677
3678  011420
3679  011420  005337  012526              DEC     TMP4
3680  011424
3681  011424                        $234:
3682  011424
3683  011424  000002                      RTI
3684
3685  011426
3686  011426                        X3SRV:
3687  011426
3688  011426  013737  001260  012502      MOV     RCSR,TMP
3689  011434
3690  011434  062737  000036  012502      ADD     #36,TMP
3691  011442
3692  011442  013777  012520  001032      MOV     TMP3,@TMP
3693  011450
3694  011450  005237  012520              INC     TMP3
3695
3696  011454
3697  011454  162737  000002  012502      SUB     #2,TMP
3698
3699
3700  011462
3701  011462  042777  000100  001012      BIC     #BIT6,@TMP
3702  011470
3703  011470  000002                      RTI
3704  011472                        SEREND:
```

```
                    ;ALL DONE
                    LET TMP4 := TMP4 - #1
                ENDIF
            ENDSRV

            BGNSRV X3SRV

                LET TMP := RCSR

                LET TMP :  TMP + #36

                LET @TMP :- TMP3

                LET TMP3 := TMP3 + #1

                ;GO BACK TO XCSR
                LET TMP := TMP - #2

                ;DISABLE XMIT INTERRUPTS,
                ;NOT EXER DOUBLE BUFF
                LET @TMP := @TMP CLR.BY #BIT6
            ENDSRV
```

```
                                        ;** RECEIVER STATUS TABLES **
                                                                ;7 BIT WDS      8 BIT WDS
011472  000100                CH0TAB: .BLKW 100                 ;  0- 37        0- 77
011672  000100                CH1TAB: .BLKW 100                 ; 40- 77      100-177
012072  000100                CH2TAB: .BLKW 100                 ;100-137      200-277
012272  000100                CH3TAB: .BLKW 100                 ;140-177      300-377
012472                      STATEND:
012472  000000              CHMSK:  .WORD 0                     ;MASK OF ACTIVE CHANNELS
012474  000000              CHCTR:  .WORD 0                     ;CHANNEL CTR
012476  000000              ACTCH:  .WORD 0                     ;ACTIVE CHANNEL CTR
012500  000000              ERWRD:  .WORD 0                     ;ERROR WORD, LOW BYTE IS NO. OF DATA ERRORS
                                                                ;HIGH BYTE INDICATES ERROR FLAG DETECTED
012502  000000              TMP:    .WORD 0                     ;TEMP STORAGE FOR SERVICE ROUTINES
012504  000000              TMP0:   0                           ;START WORD FOR CH 0
012506  000000              TMP0E:  0                           ;END   WORD FOR CH 0
012510  000000              TMP1:   0                           ;START             1
012512  000000              TMP1E:  0                           ;END               1
012514  000000              TMP2:   0                           ;START             2
012516  000000              TMP2E:  0                           ;END               2
012520  000000              TMP3:   0                           ;START             3
012522  000002              TMP3E:  .BLKW 2                     ;END               3
012526  000000              TMP4:   .WORD
012530  011472              TABL6:  CH0TAB
012532  011672                      CH1TAB
012534  012072                      CH2TAB
012536  012272                      CH3TAB
012540  000040              TABL7:  40                          ;7 BIT WORD TABLE
012542  000100                      100
012544  000140                      140
012546  000200                      200
012550  000100              TABL8:  100                         ;8 BIT WORD TABLE
012552  000200                      200
012554  000300                      300
012556  000400                      400
```

```
                        ;;*****************************************************************
                        ;* ROUTINE:TIMER
                        ;*       THIS ROUTINE IS USED TO TEST THE STATUS OF ANY BIT
                        ;*       IN ANY REGISTER.
                        ;*       INPUTS:
                        ;*            HOWLONG    THE MAXIMUM AMOUNT OF TIME TO SPEND IN
                        ;*                       THIS ROUTINE.
                        ;*            WHICHBIT     A MASK WITH THE BIT(S) SET THAT ARE
                        ;*                         TO BE CHECKED.
                        ;*            REG      A POINTER TO THE REGISTER TO BE CHECKED
                        ;*            SETCLR   THE DESIRED RESULTS
                        ;*                     EITHER #SET OR #CLEAR
                        ;*       OUTPUT:
                        ;*            THE 'C' BIT IS SET TO INDICATE AN ERROR
                        ;*            BUT IT IS TESTED BY THE IF.ERROR STATEMENT
                        ;*
                        ;*
                        ;*            NOTE::  THE USE OF (R5) IS PART OF THE LINKAGE
                        ;*                    MECHANISM BETWEEN THE CALLER AND THE CALLED
                        ;*****************************************************************

                                                ROUTINE TIMER <HOWLONG,WHICHBIT,REG,SETCLR>
012560                  TIMER:
                                                        LET     REGSAV := REG(R5) ; GET POINTER TO REGIST
012560  016537  000004  012720         MOV     REG(R5),REGSAV
012566
                                                LET     TIMSAV := HOWLONG(R5) ; SAVE HOWLONG FOR
012566  016537  000000  012722         MOV     HOWLONG(R5),TIMSAV
012574
                                                LET     FLAG :B= #0 ; INITIALIZE THE EXIT FLAG
012574  105037  012724                 CLRB    FLAG


                                        ;       START OF AN INFINITE LOOP

012600                                  LOOP
                                                 ; TEST TO SEE IF WHICHBIT IS SET
                                            IF      WHICHBIT(R5)  NOTSETIN @REGSAV THEN
012600  036577  000002  000112         BIT     WHICHBIT(R5),@REGSAV
012606  001003                         BNE     $241
                                                LET     HOLDSC :B= #0
012610  105037  012725                 CLRB    HOLDSC
012614                                  ELSE
012614  000403                         BR      $242
012616                  $241:
                                                LET     HOLDSC :B= #SET ; REMEMBER THIS
012616  112737  177777  012725         MOVB    #SET,HOLDSC
012624                                  ENDIF
012624                  $242:
                                            ; NOW SEE IF THAT WAS WHAT WE WANTED
                                            IFB     HOLDSC EQ SETCLR(R5) THEN
012624  123765  012725  000006         CMPB    HOLDSC,SETCLR(R5)
012632  001003                         BNE     $243
                                         ; JUST THE THING WE NEEDED
```

```
3804  012634
3805  012634  112737  000001  012724          MOVB    #TRUE,FLAG                      LET     FLAG :B= #TRUE
3806  012642                                                                        ENDIF
3807  012642                          $243:
3808
3809  012642                                                                    EXIFB   FLAG EQ #TRUE OR TIMSAV LE #0
3810  012642  123727  012724  000001          CMPB    FLAG,#TRUE
3811  012650  001414                          BEQ     $240
3812  012652  005737  012722                  TST     TIMSAV
3813  012656  003411                          BLE     $240
3814                                                                        ; ONE WAY OR THE OTHER, WE ARE DONE
3815                                                                    ; IF WE ARE STILL HERE THEN HANG AROUND A WHILE
3816
3817  012660                                                                    WAITMS  1       ;WAIT FOR 1 MILLI-SECONDS
3818  012660  010546                          MOV     R5,-(SP)
3819  012662  012745  000001                  MOV     #1,-(R5)
3820  012666  004737  012732                  JSR     PC,WAIT
3821  012672  012605                          MOV     (SP)+,R5
3822  012674
3823  012674  005337  012722                  DEC     TIMSAV                          LET     TIMSAV := TIMSAV - #1 ; COUNTING DOWN
3824  012700                                                                    ENDLOOP                 ; CONTINUED AT THE TOP
3825  012700  000737                          BR      $237
3826  012702                          $240:
3827
3828                                  ; ONLY 2 WAYS TO GET HERE
3829                                  ;       1). WE RAN OUT OF TIME---ERROR ! !
3830                                  ;       2). THE BIT IS IN THE CORRECT CONDITION--GOOD ! !
3831
3832  012702                                                                    IFB     FLAG EQ #TRUE THEN
3833  012702  123727  012724  000001          CMPB    FLAG,#TRUE
3834  012710  001001                          BNE     $244
3835  012712                                                                        RETURN  NO.ERROR    ; GOOD
3836  012712  000405                          BR      $235
3837  012714                                                                    ENDIF
3838  012714                          $244:
3839  012714                                                                    RETURN  ERROR           ; BAD
3840  012714  000261                          SEC
3841  012716  000404                          BR      $236
3842
3843  012720  000000                  REGSAV: .WORD 0
3844  012722  000000                  TIMSAV: .WORD 0
3845  012724     000                  FLAG:   .BYTE 0
3846  012725     000                  HOLDSC: .BYTE 0
3847                                                                        ; WE ARE DONE  GO BACK HOME
3848  012726                                                                        ENDRTN
3849  012726                          $235:
3850  012726  000241                          CLC
3851  012730                          $236:
3852  012730  000207                          RTS     PC
```

```
 3853
 3854                           ;;*****************************************************************
 3855                           ;* ROUTINE:WAIT
 3856                           ;*       THIS ROUTINE IS USED TO DELAY EXECUTION OF THE
 3857                           ;*       MAIN  PROGRAM  FOR A SPECIFIED AMOUNT OF TIME.
 3858                           ;*       THIS   IS   ACCOMPLISHED   BY  INCREMENTING  A
 3859                           ;*       REGISTER UP TO A LIMIT.  THE INNER LOOP IS SET
 3860                           ;*       TO APPROXIMATE 1 MILLI SEC.
 3861                           ;;*****************************************************************
 3862  012732                                    ROUTINE WAIT    <TIME>
 3863  012732                   WAIT:
 3864  012732                                            PUSH    <R1,R2,R3>
 3865  012740                                            LET R1 := TIME(R5)
 3866  012740  016501  000000           MOV     TIME(R5),R1
 3867  012744                                            INCRU R2 FROM #1 TO R1 BY #1
 3868  012744  012702  000001           MOV     #1,R2
 3869  012750  000402                   BR      $247
 3870  012752                   $250:
 3871  012752  062702  000001           ADD     #01,R2
 3872  012756                   $247:
 3873  012756  020201                   CMP     R2,R1
 3874  012760  101010                   BHI     $251
 3875  012762                                                    INCR R3 FROM #0 TO #100 BY #1
 3876  012762  005003                   CLR     R3
 3877  012764  000401                   BR      $252
 3878  012766                   $253:
 3879  012766  005203                   INC     R3
 3880  012770                   $252:
 3881  012770  020327  000100           CMP     R3,#100
 3882  012774  003001                   BGT     $254
 3883  012776                                                            ENDINC
 3884  012776  000773                   BR      $253
 3885  013000                   $254:
 3886  013000                                            ENDINC
 3887  013000  000764                   BR      $250
 3888  013002                   $251:
 3889  013002                                            POP     <R3,R2,R1>
 3890  013010                                    ENDRTN
 3891  013010                   $245:
 3892  013010                   $246:
 3893  013010  000207                   RTS     PC
```

```
                                        .SBTTL  ROUTINE TO SIZE THE BUS

013012  005737  001252          SIZE:   TST     $DEVM           ;BR IF INITIALLY SET
013016  001070                          BNE     4$              ;BYPASS AUTO-SIZING

013020                          1$:                             SETVEC  #ILLMEM,#INTSRV,#PR7
013020  010146                          MOV     R1,-(SP)
013022  012701  000004                  MOV     #ILLMEM,R1
013026  012721  013742                  MOV     #INTSRV,(R1)+
013032  012711  000340                  MOV     #PR7,(R1)
013036  012601                          MOV     (SP)+,R1

013040  013737  001250  013646          MOV     $BASE,TMP6
013046  012737  000001  013644          MOV     #1,SFTREG

013054  005037  013750          2$:     CLR     INTFLAG
013060  005777  000562                  TST     @TMP6           ;SEE IF THERE
013064  005737  013750                  TST     INTFLAG         ;GOT INTERRUPT?
013070  001003                          BNE     3$              ;BR IF YES  (NO DEVICE)
013072  053737  013644  001252          BIS     SFTREG,$DEVM    ;ELSE THERE, SET MAP

013100  062737  000010  013646  3$:     ADD     #10,TMP6        ;GO TO NEXT CHANNEL
013106  006337  013644                  ASL     SFTREG
013112  001360                          BNE     2$              ;BR IF MORE TO GO

013114                                                          CLRVEC  ILLMEM
013114  010146                          MOV     R1,-(SP)
013116  010246                          MOV     R2,-(SP)
013120  012701  000004                  MOV     #ILLMEM,R1
013124  010102                          MOV     R1,R2
013126  062702  000002                  ADD     #2,R2
013132  010221                          MOV     R2,(R1)+
013134  005011                          CLR     (R1)
013136  012602                          MOV     (SP)+,R2
013140  012601                          MOV     (SP)+,R1

013142  005737  001252                  TST     $DEVM           ;ANYONE THERE?
013146  001014                          BNE     4$              ;BR IF YES
013150                                  TYPTXT  <<CRLF>!NO DEV TO TEST!>
013176  000777                          BR      .               ;LOOP HERE TILL PROBLEM FIXED
                                                                ;& RESTART AT 200
013200  105737  001221          4$:     TSTB    $USWR+1         ;SEE IF ANY CONSOLE
013204  001434                          BEQ     6$              ;BR IF NONE

013206  010046                          MOV     R0,-(SP)        ;SAVE R0
013210  113700  001221                  MOVB    $USWR+1,R0      ;GET CONSOLE DEVICE X2
013214  020027  000010                  CMP     R0,#10
013220  003005                          BGT     5$              ;BR IF TOO HI
013222  056037  013650  001252          BIS     TABL1-2(R0),$DEVM       ;SET CONSOLE BIT ON MAP
013230  012600                          MOV     (SP)+,R0        ;RESTORE R0
013232  000421                          BR      6$

013234                          5$:     TYPTXT  <<CRLF>!CONSOLE IN $USWR WRONG!>
013272  000137  014514                  JMP     $EOP                    ;START ALL OVER
```

```
3950                                                    ;STACK WILL RESET IN 'LOOP'
3951  013276  013737  001252  013634  6$:     MOV     $DEVM,MASK1
3952  013304  013737  001252  013636          MOV     $DEVM,MASK2
3953  013312  013737  001252  013640          MOV     $DEVM,MASK3
3954  013320  013737  001252  013642          MOV     $DEVM,MASK4
3955
3956  013326  042737  177760  013634          BIC     # C<17>,MASK1           ;SAVE ONLY MODULE 1 CHANNELS
3957  013334  042737  177417  013636          BIC     # C<360>,MASK2          ;                 2
3958  013342  042737  170377  013640          BIC     # C<7400>,MASK3         ;                 3
3959  013350  042737  007777  013642          BIC     # C<170000>,MASK4       ;                 4
3960
3961  013356  006237  013636                  ASR     MASK2
3962  013362  006237  013636                  ASR     MASK2
3963  013366  006237  013636                  ASR     MASK2
3964  013372  006237  013636                  ASR     MASK2                   ;RIGHT JUSTIFY MODULE 2 CHANNELS
3965
3966  013376  000337  013640                  SWAB    MASK3                   ;                     3
3967
3968  013402  000337  013642                  SWAB    MASK4
3969  013406  006237  013642                  ASR     MASK4
3970  013412  006237  013642                  ASR     MASK4
3971  013416  006237  013642                  ASR     MASK4
3972  013422  006237  013642                  ASR     MASK4                   ;                     4
3973
3974  013426                                  TYPTXT  <<CRLF>!WILL TEST:!<CRLF>>
3975
3976  013452  012700  013634                  MOV     #MASK1,R0
3977  013456  012737  000061  013626          MOV     #61,MODNUM              ;INIT MOD #
3978  013464  012737  000060  013632          MOV     #60,CHNUM+2             ;INIT CHANNEL #
3979
3980  013472  012037  013650          7$:     MOV     (R0)+,MSK
3981  013476  001001                          BNE     8$                      ;BR IF ANY THERE
3982  013500  000441                          BR      11$                     ;ELSE TRY NEXT MODULE
3983
3984  013502                          8$:     TYPTXT  <<CRLF>!MODULE !>
3985  013522  104401  013626                  TYPE    ,MODNUM
3986  013526                                  TYPTXT  <!  CHANNEL !>
3987
3988  013550  032737  000001  013650  9$:     BIT     #BIT0,MSK
3989  013556  001402                          BEQ     10$                     ;BR IF NOT TESTING THAT CHANNEL
3990  013560  104401  013630                  TYPE    ,CHNUM                  ;ELSE TYPE IT
3991
3992  013564  005237  013632          10$:    INC     CHNUM+2                 ;BUMP CHANNEL CTR
3993  013570  006237  013650                  ASR     MSK
3994  013574  001365                          BNE     9$                      ;BR IF MORE TO CHECK
3995  013576  012737  000060  013632          MOV     #60,CHNUM+2             ;ELSE RESET CH #
3996
3997  013604  005237  013626          11$:    INC     MODNUM                  ;BUMP MOD #
3998  013610  023727  013626  000065          CMP     MODNUM,#65              ;ALL DONE?
3999  013616  001325                          BNE     7$                      ;BR IF NO
4000  013620  104401  001171                  TYPE    ,$CRLF
4001
4002  013624  000207                          RTS     PC                      ;ELSE EXIT
4003
4004  013626  000061                  MODNUM: .ASCIZ  /1/
4005  013630  020040  000060          CHNUM:  .ASCIZ  /  0/
```

```
 4006
 4007  013634  000000                MASK1:  0                       ;CONTAINS MODULE 1 CHANNELS
 4008  013636  000000                MASK2:  0                       ;                2
 4009  013640  000000                MASK3:  0                       ;                3
 4010  013642  000000                MASK4:  0                       ;                4
 4011
 4012  013644  000000                SFTREG: 0
 4013  013646  000000                TMP6:   0
 4014  013650  000000                MSK:    0
 4015
 4016  013652  000010                TABL1:  10                      ;SET $DEVM IF CONSOLE ON MODULE 1
 4017  013654  000200                        200                     ;                           2
 4018  013656  004000                        4000                    ;                           3
 4019  013660  100000                        100000                  ;                           4
```

```
                                .SBTTL  ROUTINE TO TEST FOR CONSOLE DEVICE

013662  113737  001221  013740  TSTCON: MOVB    $USWR+1,TMP7            ;GET CONSOLE INFO ONLY
013670  006237  013740          ASR     TMP7                    ;RIGHT JUSTIFY
013674  023737  013740  001206  CMP     TMP7,$UNIT              ;SEE IF UNIT UNDER TEST HAS CONSOLE
013702  001403                  BEQ     1$                      ;BR IF YES
013704  005037  013736          CLR     CONSOLE                 ;ELSE CLEAR STALE INFO
013710  000207                  RTS     PC

013712  013737  001144  014310  1$:     MOV     $TKS,ADDRESS            ;FORCE TTY ADDRESS
013720  012737  000060  014312          MOV     #TKVEC,VECTOR           ;       VECTOR
013726  012737  000001  013736          MOV     #TRUE,CONSOLE           ;SET FLAG
013734  000207                  RTS     PC

013736  000000          CONSOLE:        0                       ;FLAG, 1= TESTING CONSOLE DEVICE
013740  000000          TMP7:           0                       ;HOLD CONSOLE #



                                .SBTTL  INTSRV  INTERRUPT SERVICE ROUTINE

                                ;;*******************************************************************
                                ;*       THIS GLOBAL ROUTINE DOES NOTHING BUT INCREMENT
                                ;*       'INTFLAG'  EACH TIME IT IS CALLED.  IT ASSUMES
                                ;*       THAT THE MAIN CALLING ROUTINE WILL  KNOW  WHAT
                                ;*       TO LOOK FOR.
                                ;;*******************************************************************
013742                          BGNSRV INTSRV                   ;GLOBAL INTERRUPT SERVICE ROUTINE
013742                          INTSRV:
                                                        ;ADD 1 TO 'INTERRUPT OCCURED' FLAG
                                                                LET INTFLAG := INTFLAG + #1
013742  005237  013750                  INC     INTFLAG
013746                          ENDSRV                          ;THAT'S ALL
013746  000002                  RTI
013750  000000          INTFLAG:  0
```

```
4057
4058                                    .SBTTL  ROUTINE TO SET UP FOR NEXT MODULE/CHANNEL ADDRESS
4059
4060
4061                                    ;;*****************************************************************
4062                                    ;*      THIS  ROUTINE  CAUSES  ADRS   TO POINT TO  THE
4063                                    ;*      ADDRESS  OF  DLV11-J  UNDER  TEST,  ADRS +2 TO
4064                                    ;*      POINT TO THE VECTOR OF THE DLV11-J UNDER TEST.
4065                                    ;;*****************************************************************
4066
4067  013752  013737  001250  014310  CYCLE:  MOV     $BASE,ADDRESS
4068  013760  013737  001244  014312          MOV     $VECT1,VECTOR
4069
4070  013766  023727  001206  000005          CMP     $UNIT,#5                ;DONE ALL MODULES?
4071  013774  001013                          BNE     1$                      ;BR IF NO
4072  013776  005037  014320                  CLR     PHASE2                  ;RE-INIT
4073  014002  005037  014316                  CLR     CHAN
4074  014006  012737  000001  001206          MOV     #1,$UNIT
4075  014014  005037  014322                  CLR     P1CNT
4076  014020  000137  014514                  JMP     $EOP                    ;& EXIT BACK TO 'LOOP'
4077                                                                          ;STACK WILL GET RESET THERE.
4078  014024  013700  001206          1$:     MOV     $UNIT,R0
4079  014030  006300                          ASL     R0                      ;MULT BY 2 TO INDEX INTO TABLES
4080  014032  066037  014246  014310          ADD     TABL2-2(R0),ADDRESS     ;ADD MODULE OFFSET
4081  014040  066037  014246  014312          ADD     TABL2-2(R0),VECTOR      ;ADD VECTOR OFFSET
4082  014046  017037  014256  014314          MOV     @TABL3-2(R0),MASK       ;STORE CHANNELS IN 'MASK'
4083
4084  014054  005037  013736                  CLR     CONSOLE
4085  014060  005737  014314                  TST     MASK                    ;ANY CHANNELS TO BE TESTED?
4086  014064  001004                          BNE     2$                      ;BR IF YES
4087  014066  005237  001206                  INC     $UNIT                   ;ELSE BUMP UNIT #
4088  014072  000137  013752                  JMP     CYCLE                   ;TRY AGAIN
4089
4090  014076  023727  014320  000001  2$:     CMP     PHASE2,#TRUE            ;DOING PHASE 2?
4091  014104  001452                          BEQ     6$                      ;BR IF YES
4092  014106  023727  014316  000004          CMP     CHAN,#4                 ;ELSE, DONE ALL CHANNELS?
4093  014114  001021                          BNE     4$                      ;BR IF NO
4094
4095  014116  005037  014316                  CLR     CHAN                    ;ELSE BACK TO 0
4096  014122  032737  000020  001220          BIT     #WRAP,$USWR             ;DOING DATA WRAP TESTS?
4097  014130  001004                          BNE     3$                      ;BR IF YES
4098  014132  005237  001206                  INC     $UNIT                   ;ELSE GO TO NEXT UNIT, NO PHASE 2
4099  014136  000137  013752                  JMP     CYCLE                   ;& DO IT
4100
4101  014142  012737  000001  014320  3$:     MOV     #TRUE,PHASE2
4102  014150  005037  014322                  CLR     P1CNT
4103  014154  000137  013752                  JMP     CYCLE                   ;GO TO PHASE 2 TESTING
4104
4105  014160  013700  014316          4$:     MOV     CHAN,R0
4106  014164  006300                          ASL     R0                      ;MULT BY 2 TO INDEX TABLE
4107  014166  066037  014270  014310          ADD     TABL4(R0),ADDRESS       ;ADD CHANNEL OFFSET
4108  014174  066037  014270  014312          ADD     TABL4(R0),VECTOR        ;ADD VECTOR OFFSET
4109  014202  036037  014300  014314          BIT     TABL5(R0),MASK          ;IS THAT CHANNEL TO BE TESTED?
4110  014210  001413                          BEQ     7$                      ;BR IF NO
4111
4112  014212  023727  014316  000003          CMP     CHAN,#3                 ;IS IT CHAN 3?
```

```
 4113  014220  001002                    BNE     5$                      ;BR IF NO
 4114  014222  004737  013662            JSR     PC,TSTCON               ;ELSE SEE IF ITS THE CONSOLE
 4115
 4116  014226  005237  014316  5$:       INC     CHAN                    ;BUMP CHANNEL CTR FOR NEXT TIME
 4117  014232  012701  014310  6$:       MOV     #ADDRESS,ADRS
 4118  014236  000207                    RTS     PC                      ;GO TO PHASE 1 TESTS
 4119
 4120  014240  005237  014316  7$:       INC     CHAN
 4121  014244  000137  013752            JMP     CYCLE                   ;TRY AGAIN
 4122
 4123  014250  000000            TABL2:  0               ;OFFSET FOR MODULE 1
 4124  014252  000040                    40              ;                  2
 4125  014254  000100                    100             ;                  3
 4126  014256  000140                    140             ;                  4
 4127
 4128  014260  013634            TABL3:  MASK1           ;MODULE 1 CHANNEL DATA
 4129  014262  013636                    MASK2           ;        2
 4130  014264  013640                    MASK3           ;        3
 4131  014266  013642                    MASK4           ;        4
 4132
 4133  014270  000000            TABL4:  0               ;CHANNEL 0 OFFSET
 4134  014272  000010                    10              ;         1
 4135  014274  000020                    20              ;         2
 4136  014276  000030                    30              ;         3
 4137
 4138  014300  000001            TABL5:  BIT0            ;BIT TO TEST CHANNEL 0 PRESENT
 4139  014302  000002                    BIT1            ;                    1
 4140  014304  000004                    BIT2            ;                    2
 4141  014306  000010                    BIT3            ;                    3
 4142
 4143  014310  000000            ADDRESS: 0
 4144  014312  000000            VECTOR: 0
 4145  014314  000000            MASK:   0
 4146  014316  000000            CHAN:   0
 4147  014320  000000            PHASE2: 0
 4148  014322  000000            P1CNT:  0               ;CTR TO PRINT 'PHASE 1' ONLY ONCE
 4149
 4150
```

```
 4151
 4152  014324                                                ROUTINE MYTYPE
 4153  014324                          MYTYPE:
 4154  014324                                  TYPTXT  <<CRLF>*TEST # *>
 4155  014344                                  TYPOCT  $TESTN
 4156  014352                                  TYPTXT  <*,ERROR # *>
 4157  014372  113737  001114  001176          MOVB    $ITEMB,$FATAL   ; APT FATAL ERROR NUMBER
 4158  014400                                  TYPOCS  $FATAL
 4159  014410                                  TYPTXT  <*,PC = *>
 4160  014426                                  TYPOCT  $ERRPC
 4161  014434                                  TYPTXT  <*,CSR: *>
 4162  014452                                  TYPOCT  DLADD
 4163  014460                                  TYPTXT  <*,VECTOR: *>
 4164  014500                                  TYPOCT  DLVEC
 4165  014506  104401  001171                  TYPE    ,$CRLF
 4166  014512                                                  ENDRTN
 4167  014512                          $255:
 4168  014512                          $256:
 4169  014512  000207                          RTS     PC
 4170
```

```
                        .SBTTL  END OF PASS ROUTINE

                        ;;**********************************************************************
                        ;*INCREMENT THE PASS NUMBER ($PASS)
                        ;*INDICATE END-OF-PROGRAM AFTER 1 PASSES THRU THE PROGRAM
                        ;*TYPE "END PASS #XXXXX" (WHERE XXXXX IS A DECIMAL NUMBER)
                        ;*IF THERES A MONITOR GO TO IT
                        ;*IF THERE ISN'T JUMP TO LOOP

014514                  $EOP:
014514  000004                  SCOPE
014516  005037  001102          CLR     $TSTNM          ;;ZERO THE TEST NUMBER
014522  005037  001160          CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
014526  005237  001202          INC     $PASS           ;;INCREMENT THE PASS NUMBER
014532  042737  100000  001202  BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
014540  005327                  DEC     (PC)+           ;;LOOP?
014542  000001          $EOPCT: .WORD   1
014544  003022                  BGT     $DOAGN          ;;YES
014546  012737                  MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
014550  000001          $ENDCT: .WORD   1
014552  014542                  $EOPCT
014554  104401  014621          TYPE    ,$ENDMG         ;;TYPE "END PASS #"
014560  013746  001202          MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
014564  104405                  TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
014566  104401  014616          TYPE    ,$ENULL         ;;TYPE A NULL CHARACTER
014572  013700  000042  $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
014576  001405                  BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
014600  000005                  RESET                   ;;CLEAR THE WORLD
014602  004710          $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
014604  000240                  NOP                     ;;SAVE ROOM
014606  000240                  NOP                     ;;FOR
014610  000240                  NOP                     ;;ACT11
014612                  $DOAGN:
014612  000137                  JMP     @(PC)+          ;;RETURN
014614  001732          $RTNAD: .WORD   LOOP
014616     377     377     000  $ENULL: .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
014621     015  042412  042116  $ENDMG: .ASCIZ  <15><12>/END PASS #/
014626  050040  051501  020123
014634  000043
```

```
                                        .SBTTL  TYPE ROUTINE
                                        ;;*********************************************************************
                                        ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
                                        ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
                                        ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
                                        ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
                                        ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
                                        ;*
                                        ;*CALL:
                                        ;*1) USING A TRAP INSTRUCTION
                                        ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
                                        ;*OR
                                        ;*      TYPE
                                        ;*      MESADR
                                        ;*

014636  105737  001157          $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
014642  100002                          BPL     1$              ;;BR IF YES
014644  000000                          HALT                    ;;HALT HERE IF NO TERMINAL
014646  000430                          BR      3$              ;;LEAVE
014650  010046                  1$:     MOV     R0,-(SP)        ;;SAVE R0
014652  017600  000002                  MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
014656  122737  000001  001214          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
014664  001011                          BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
014666  132737  000100  001215          BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
014674  001405                          BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
014676  010037  014706                  MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
014702  004737  015702                  JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
014706  000000                  61$:    .WORD   0               ;;MESSAGE ADDRESS
014710  132737  000040  001215  62$:    BITB    #APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
014716  001003                          BNE     60$             ;;YES,SKIP TYPE OUT
014720  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
014722  001005                          BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
014724  005726                          TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
014726  012600                  60$:    MOV     (SP)+,R0        ;;RESTORE R0
014730  062716  000002          3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
014734  000002                          RTI                     ;;RETURN
014736  122716  000011          4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
014742  001430                          BEQ     8$
014744  122716  000200                  CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
014750  001006                          BNE     5$
014752  005726                          TST     (SP)+           ;;POP  <CR><LF> EQUIV
014754  104401                          TYPE                    ;;TYPE A CR AND LF
014756  001171                          $CRLF
014760  105037  015114                  CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
014764  000755                          BR      2$              ;;GET NEXT CHARACTER
014766  004737  015050          5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
014772  123726  001156          6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
014776  001350                          BNE     2$              ;;IF NO GO GET NEXT CHAR.
015000  013746  001154                  MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
                                                                ;;AND THE NULL CHAR.
015004  105366  000001          7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
015010  002770                          BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
015012  004737  015050                  JSR     PC,$TYPEC       ;;GO TYPE A NULL
015016  105337  015114                  DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
```

```
 4266  015022  000770                        BR      7$              ;;LOOP
 4267
 4268                                ;HORIZONTAL TAB PROCESSOR
 4269
 4270  015024  112716  000040          8$:   MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
 4271  015030  004737  015050          9$:   JSR     PC,$TYPEC       ;;TYPE A SPACE
 4272  015034  132737  000007  015114        BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
 4273  015042  001372                        BNE     9$              ;;TAB STOP
 4274  015044  005726                        TST     (SP)+           ;;POP SPACE OFF STACK
 4275  015046  000724                        BR      2$              ;;GET NEXT CHARACTER
 4276  015050  105777  164074          $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
 4277  015054  100375                        BPL     $TYPEC
 4278  015056  116677  000002  164066        MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
 4279  015064  122766  000015  000002        CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
 4280  015072  001003                        BNE     1$              ;;BRANCH IF NO
 4281  015074  105037  015114                CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
 4282  015100  000406                        BR      $TYPEX          ;;EXIT
 4283  015102  122766  000012  000002  1$:   CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
 4284  015110  001402                        BEQ     $TYPEX          ;;BRANCH IF YES
 4285  015112  105227                        INCB    (PC)+           ;;COUNT THE CHARACTER
 4286  015114  000000          $CHARCNT:.WORD  0               ;;CHARACTER COUNT STORAGE
 4287  015116  000207          $TYPEX: RTS     PC
 4288
```

```
                                        .SBTTL  TTY INPUT ROUTINE

                                        ;;**********************************************************************
                                        .ENABL  LSB

                                        ;;**********************************************************************
                                        ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
                                        ;*ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
                                        ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP CALL
                                        ;*WHEN OPERATING IN TTY FLAG MODE.
015120  022737  000176  001140  $CKSWR: CMP     #SWREG,SWR      ;;IS THE SOFT-SWR SELECTED?
015126  001074                        BNE     15$             ;;BRANCH IF NO
015130  105777  164010                TSTB    @$TKS           ;;CHAR THERE?
015134  100071                        BPL     15$             ;;IF NO, DON'T WAIT AROUND
015136  117746  164004                MOVB    @$TKB,-(SP)     ;;SAVE THE CHAR
015142  042716  177600                BIC     # C177,(SP)     ;;STRIP-OFF THE ASCII
015146  022726  000007                CMP     #7,(SP)+        ;;IS IT A CONTROL G?
015152  001062                        BNE     15$             ;;NO, RETURN TO USER
015154  123727  001134  000001        CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
015162  001456                        BEQ     15$             ;;BRANCH IF YES

015164  104401  015645                TYPE    ,$CNTLG         ;;ECHO THE CONTROL-G ( G)
015170  104401  015652        $GTSWR: TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
015174  013746  000176                MOV     SWREG,-(SP)     ;;SAVE SWREG FOR TYPEOUT
015200  104402                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
015202  104401  015663                TYPE    ,$MNEW          ;;PROMPT FOR NEW SWR
015206  005046                19$:    CLR     -(SP)           ;;CLEAR COUNTER
015210  005046                        CLR     -(SP)           ;;THE NEW SWR
015212  105777  163726        7$:     TSTB    @$TKS           ;;CHAR THERE?
015216  100375                        BPL     7$              ;;IF NOT TRY AGAIN

015220  117746  163722                MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
015224  042716  177600                BIC     # C177,(SP)     ;;MAKE IT 7-BIT ASCII


015230  021627  000025        9$:     CMP     (SP),#25        ;;IS IT A CONTROL-U?
015234  001005                        BNE     10$             ;;BRANCH IF NOT
015236  104401  015640                TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U ( U)
015242  062706  000006        20$:    ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
015246  000757                        BR      19$             ;;LET'S TRY IT AGAIN


015250  021627  000015        10$:    CMP     (SP),#15        ;;IS IT A <CR>?
015254  001022                        BNE     16$             ;;BRANCH IF NO
015256  005766  000004                TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
015262  001403                        BEQ     11$             ;;BRANCH IF YES
015264  016677  000002  163646        MOV     2(SP),@SWR      ;;SAVE NEW SWR
015272  062706  000006        11$:    ADD     #6,SP           ;;CLEAR UP STACK
015276  104401  001171        14$:    TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
015302  123727  001135  000001        CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
015310  001003                        BNE     15$             ;;BRANCH IF NOT
015312  012777  000100  163624        MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
015320  000002                15$:    RTI                     ;;RETURN
015322  004737  015050        16$:    JSR     PC,$TYPEC       ;;ECHO CHAR
015326  021627  000060                CMP     (SP),#60        ;;CHAR < 0?
```

```
 4345  015332  002420                        BLT     18$             ;;BRANCH IF YES
 4346  015334  021627  000067                CMP     (SP),#67        ;;CHAR > 7?
 4347  015340  003015                        BGT     18$             ;;BRANCH IF YES
 4348  015342  042726  000060                BIC     #60,(SP)+       ;;STRIP-OFF ASCII
 4349  015346  005766  000002                TST     2(SP)           ;;IS THIS THE FIRST CHAR
 4350  015352  001403                        BEQ     17$             ;;BRANCH IF YES
 4351  015354  006316                        ASL     (SP)            ;;NO, SHIFT PRESENT
 4352  015356  006316                        ASL     (SP)            ;;   CHAR OVER TO MAKE
 4353  015360  006316                        ASL     (SP)            ;;   ROOM FOR NEW ONE.
 4354  015362  005266  000002        17$:    INC     2(SP)           ;;KEEP COUNT OF CHAR
 4355  015366  056616  177776                BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
 4356  015372  000707                        BR      7$              ;;GET THE NEXT ONE
 4357  015374  104401  001170        18$:    TYPE    ,$QUES          ;;TYPE ?<CR><LF>
 4358  015400  000720                        BR      20$             ;;SIMULATE CONTROL-U
 4359                                .DSABL  LSB
 4360
 4361
 4362                                ;;*****************************************************************
 4363                                ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
 4364                                ;*CALL:
 4365                                ;*      RDCHR                   ;;INPUT A SINGLE CHARACTER FROM THE TTY
 4366                                ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
 4367                                ;*                              ;;WITH PARITY BIT STRIPPED OFF
 4368                                ;
 4369
 4370  015402  011646                $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC
 4371  015404  016666  000004  000002        MOV     4(SP),2(SP)     ;;SAVE THE PS
 4372  015412  105777  163526        1$:     TSTB    @$TKS           ;;WAIT FOR
 4373  015416  100375                        BPL     1$              ;;A CHARACTER
 4374  015420  117766  163522  000004        MOVB    @$TKB,4(SP)     ;;READ THE TTY
 4375  015426  042766  177600  000004        BIC     # C<177>,4(SP)  ;;GET RID OF JUNK IF ANY
 4376  015434  026627  000004  000023        CMP     4(SP),#23       ;;IS IT A CONTROL-S?
 4377  015442  001013                        BNE     3$              ;;BRANCH IF NO
 4378  015444  105777  163474        2$:     TSTB    @$TKS           ;;WAIT FOR A CHARACTER
 4379  015450  100375                        BPL     2$              ;;LOOP UNTIL ITS THERE
 4380  015452  117746  163470                MOVB    @$TKB,-(SP)     ;;GET CHARACTER
 4381  015456  042716  177600                BIC     # C177,(SP)     ;;MAKE IT 7-BIT ASCII
 4382  015462  022627  000021                CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
 4383  015466  001366                        BNE     2$              ;;IF NOT DISCARD IT
 4384  015470  000750                        BR      1$              ;;YES, RESUME
 4385  015472  026627  000004  000140  3$:   CMP     4(SP),#140      ;;IS IT UPPER CASE?
 4386  015500  002407                        BLT     4$              ;;BRANCH IF YES
 4387  015502  026627  000004  000175        CMP     4(SP),#175      ;;IS IT A SPECIAL CHAR?
 4388  015510  003003                        BGT     4$              ;;BRANCH IF YES
 4389  015512  042766  000040  000004        BIC     #40,4(SP)       ;;MAKE IT UPPER CASE
 4390  015520  000002                4$:     RTI                     ;;GO BACK TO USER
 4391                                ;;*****************************************************************
 4392                                ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
 4393                                ;*CALL:
 4394                                ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
 4395                                ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
 4396                                ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
 4397
 4398  015522  010346                $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
 4399  015524  012703  015630        1$:     MOV     #$TTYIN,R3      ;;GET ADDRESS
 4400  015530  022703  015640        2$:     CMP     #$TTYIN+8.,R3   ;;BUFFER FULL?
```

```
 4401  015534  101405                    BLOS    4$              ;;BR IF YES
 4402  015536  104410                    RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
 4403  015540  112613                    MOVB    (SP)+,(R3)      ;;GET CHARACTER
 4404  015542  122713  000177      10$:  CMPB    #177,(R3)       ;;IS IT A RUBOUT
 4405  015546  001003                    BNE     3$              ;;SKIP IF NOT
 4406  015550  104401  001170      4$:   TYPE    ,$QUES          ;;TYPE A '?'
 4407  015554  000763                    BR      1$              ;;CLEAR THE BUFFER AND LOOP
 4408  015556  111337  015626      3$:   MOVB    (R3),9$         ;;ECHO THE CHARACTER
 4409  015562  104401  015626            TYPE    ,9$
 4410  015566  122723  000015            CMPB    #15,(R3)+       ;;CHECK FOR RETURN
 4411  015572  001356                    BNE     2$              ;;LOOP IF NOT RETURN
 4412  015574  105063  177777            CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
 4413  015600  104401  001172            TYPE    ,$LF            ;;TYPE A LINE FEED
 4414  015604  012603                    MOV     (SP)+,R3        ;;RESTORE R3
 4415  015606  011646                    MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
 4416  015610  016666  000004  000002    MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
 4417  015616  012766  015630  000004    MOV     #$TTYIN,4(SP)
 4418  015624  000002                    RTI                     ;;RETURN
 4419  015626     000            9$:    .BYTE   0               ;;STORAGE FOR ASCII CHAR. TO TYPE
 4420  015627     000                    .BYTE   0               ;;TERMINATOR
 4421  015630  000010         $TTYIN:   .BLKB   8.              ;;RESERVE 8 BYTES FOR TTY INPUT
 4422  015640  052536  005015     000   $CNTLU:   .ASCIZ  / U/<15><12>    ;;CONTROL "U"
 4423  015645     136  006507  000012   $CNTLG:   .ASCIZ  / G/<15><12>    ;;CONTROL "G"
 4424  015652  005015  053523  020122   $MSWR:    .ASCIZ  <15><12>/SWR = /
 4425  015660  020075     000
 4426  015663     040  047040  053505   $MNEW:    .ASCIZ  /  NEW = /
 4427  015670  036440  000040
```

```
                                        .SBTTL  APT COMMUNICATIONS ROUTINE
                                        ;;*****************************************************************
015674  112737  000001  016140  $ATY1:  MOVB    #1,$FFLG        ;;TO REPORT FATAL ERROR
015702  112737  000001  016136  $ATY3:  MOVB    #1,$MFLG        ;;TO TYPE A MESSAGE
015710  000403                  BR      $ATYC
015712  112737  000001  016140  $ATY4:  MOVB    #1,$FFLG        ;;TO ONLY REPORT FATAL ERROR
015720                          $ATYC:
015720  010046                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
015722  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
015724  105737  016136          TSTB    $MFLG           ;;SHOULD TYPE A MESSAGE?
015730  001450                  BEQ     5$              ;;IF NOT:  BR
015732  122737  000001  001214  CMPB    #APTENV,$ENV    ;;OPERATING UNDER APT?
015740  001031                  BNE     3$              ;;IF NOT:  BR
015742  132737  000100  001215  BITB    #APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
015750  001425                  BEQ     3$              ;;IF NOT:  BR
015752  017600  000004          MOV     @4(SP),R0       ;;GET MESSAGE ADDR.
015756  062766  000002  000004  ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
015764  005737  001174          1$:     TST     $MSGTYPE        ;;SEE IF DONE W/ LAST XMISSION?
015770  001375                  BNE     1$              ;;IF NOT:  WAIT
015772  010037  001210          MOV     R0,$MSGAD       ;;PUT ADDR IN MAILBOX
015776  105720                  2$:     TSTB    (R0)+           ;;FIND END OF MESSAGE
016000  001376                  BNE     2$
016002  163700  001210          SUB     $MSGAD,R0       ;;SUB START OF MESSAGE
016006  006200                  ASR     R0              ;;GET MESSAGE LNGTH IN WORDS
016010  010037  001212          MOV     R0,$MSGLGT      ;;PUT LENGTH IN MAILBOX
016014  012737  000004  001174  MOV     #4,$MSGTYPE     ;;TELL APT TO TAKE MSG.
016022  000413                  BR      5$
016024  017637  000004  016050  3$:     MOV     @4(SP),4$       ;;PUT MSG ADDR IN JSR LINKAGE
016032  062766  000002  000004  ADD     #2,4(SP)                ;;BUMP RETURN ADDRESS
016040  013746  177776          MOV     177776,-(SP)    ;;PUSH 177776 ON STACK
016044  004737  014636          JSR     PC,$TYPE        ;;CALL TYPE MACRO
016050  000000                  4$:     .WORD   0
016052                          5$:
016052  105737  016140          10$:    TSTB    $FFLG           ;;SHOULD REPORT FATAL ERROR?
016056  001416                  BEQ     12$             ;;IF NOT:  BR
016060  005737  001214          TST     $ENV            ;;RUNNING UNDER APT?
016064  001413                  BEQ     12$             ;;IF NOT:  BR
016066  005737  001174          11$:    TST     $MSGTYPE        ;;FINISHED LAST MESSAGE?
016072  001375                  BNE     11$             ;;IF NOT:  WAIT
016074  017637  000004  001176  MOV     @4(SP),$FATAL   ;;GET ERROR #
016102  062766  000002  000004  ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
016110  005237  001174          INC     $MSGTYPE        ;;TELL APT TO TAKE ERROR
016114  105037  016140          12$:    CLRB    $FFLG           ;;CLEAR FATAL FLAG
016120  105037  016137          CLRB    $LFLG           ;;CLEAR LOG FLAG
016124  105037  016136          CLRB    $MFLG           ;;CLEAR MESSAGE FLAG
016130  012601                  MOV     (SP)+,R1        ;;POP STACK INTO R1
016132  012600                  MOV     (SP)+,R0        ;;POP STACK INTO R0
016134  000207                  RTS     PC              ;;RETURN
016136     000                  $MFLG:  .BYTE   0       ;;MESSG. FLAG
016137     000                  $LFLG:  .BYTE   0       ;;LOG FLAG
016140     000                  $FFLG:  .BYTE   0       ;;FATAL FLAG
016142                                  .EVEN
000200                  APTSIZE=200
000001                  APTENV=001
000100                  APTSPOOL=100
```

```
 4484          000040                  APTCSUP=040
```

```
                                        .SBTTL  ERROR HANDLER ROUTINE

                                        ;;**********************************************************************
                                        ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
                                        ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
                                        ;*AND GO TO MYTYPE ON ERROR
                                        ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
                                        ;*SW15=1         HALT ON ERROR
                                        ;*SW13=1         INHIBIT ERROR TYPEOUTS
                                        ;*SW10=1         BELL ON ERROR
                                        ;*SW09=1         LOOP ON ERROR
                                        ;*CALL
                                        ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER

016142                                  $ERROR:
016142  104407                                  CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
016144  105237  001103                7$:       INCB    $ERFLG          ;;SET THE ERROR FLAG
016150  001775                                  BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
016152  013777  001102  162762                MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
016160  032777  002000  162752                BIT     #BIT10,@SWR     ;;BELL ON ERROR?
016166  001402                                BEQ     1$              ;;NO - SKIP
016170  104401  001164                        TYPE    ,$BELL          ;;RING BELL
016174  005237  001112                1$:       INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
016200  011637  001116                        MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
016204  162737  000002  001116                SUB     #2,$ERRPC
016212  117737  162700  001114                MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
016220  032777  020000  162712                BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
016226  001004                                BNE     20$             ;;SKIP TYPEOUTS
016230  004737  014324                        JSR     PC,MYTYPE       ;;GO TO USER ERROR ROUTINE
016234  104401  001171                        TYPE    ,$CRLF
016240                                  20$:
016240  122737  000001  001214                CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
016246  001007                                BNE     2$              ;;NO,SKIP APT ERROR REPORT
016250  113737  001114  016262                MOVB    $ITEMB,21$      ;;SET ITEM NUMBER AS ERROR NUMBER
016256  004737  015712                        JSR     PC,$ATY4        ;;REPORT FATAL ERROR TO APT
016262     000                  21$:       .BYTE   0
016263     000                             .BYTE   0
016264  000777                  22$:       BR      22$             ;;APT ERROR LOOP
016266  005777  162646                2$:       TST     @SWR            ;;HALT ON ERROR
016272  100002                                BPL     3$              ;;SKIP IF CONTINUE
016274  000000                                HALT                    ;;HALT ON ERROR!
016276  104407                                CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
016300  032777  001000  162632          3$:       BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
016306  001402                                BEQ     4$              ;;BR IF NO
016310  013716  001110                        MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
016314  005737  001162                4$:       TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
016320  001402                                BEQ     5$              ;;BR IF NONE
016322  013716  001162                        MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
016326                                  5$:
016326  022737  014602  000042                CMP     #$ENDAD,@#42    ;;ACT-11 AUTO-ACCEPT?
016334  001001                                BNE     6$              ;;BRANCH IF NO
016336  000000                                HALT                    ;;YES
016340                                  6$:
016340  000002                                RTI                     ;;RETURN
```

```
                        .SBTTL  SCOPE HANDLER ROUTINE

                        ;;*********************************************************************
                        ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
                        ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
                        ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
                        ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
                        ;*SW14=1         LOOP ON TEST
                        ;*SW11=1         INHIBIT ITERATIONS
                        ;*SW09=1         LOOP ON ERROR
                        ;*SW08=1         LOOP ON TEST IN SWR<7:0>
                        ;*CALL
                        ;*      SCOPE           ;;SCOPE=IOT

016342                  $SCOPE:
016342  104407                 CKSWR                   ;;TEST  FOR CHANGE IN SOFT-SWR
016344  032777  040000  162566 1$:    BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
016352  001114                 BNE     $OVER           ;;YES IF SW14=1
                        ;#####START OF CODE FOR THE XOR TESTER#####
016354  000416          $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
                                                        ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
016356  013746  000004         MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
016362  012737  016402  000004 MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
016370  005737  177060         TST     @#177060        ;;TIME OUT ON XOR?
016374  012637  000004         MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
016400  000463                 BR      $SVLAD          ;;GO TO THE NEXT TEST
016402  022626          5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
016404  012637  000004         MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
016410  000423                 BR      7$              ;;LOOP ON THE PRESENT TEST
016412                  6$:;#####END OF CODE FOR THE XOR TESTER#####
016412  032777  000400  162520         BIT     #BIT08,@SWR     ;;LOOP ON SPEC. TEST?
016420  001404                 BEQ     2$              ;;BR IF NO
016422  127737  162512  001102         CMPB    @SWR,$TSTNM     ;;ON THE RIGHT TEST?    SWR<7:0>
016430  001465                 BEQ     $OVER           ;;BR IF YES
016432  105737  001103  2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
016436  001421                 BEQ     3$              ;;BR IF NO
016440  123737  001115  001103         CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
016446  101015                 BHI     3$              ;;BR IF NO
016450  032777  001000  162462         BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
016456  001404                 BEQ     4$              ;;BR IF NO
016460  013737  001110  001106 7$:     MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
016466  000446                 BR      $OVER
016470  105037  001103  4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
016474  005037  001160         CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
016500  000415                 BR      1$              ;;ESCAPE TO THE NEXT TEST
016502  032777  004000  162430 3$:     BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
016510  001011                 BNE     1$              ;;BR IF YES
016512  005737  001202         TST     $PASS           ;;IF FIRST PASS OF PROGRAM
016516  001406                 BEQ     1$              ;;      INHIBIT ITERATIONS
016520  005237  001104         INC     $ICNT           ;;INCREMENT ITERATION COUNT
016524  023737  001160  001104 CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
016532  002024                 BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
016534  012737  000001  001104 1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
016542  013737  016620  001160         MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
016550  105237  001102  $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
016554  113737  001102  001200         MOVB    $TSTNM,$TESTN   ;;SET TEST NUMBER IN APT MAILBOX
```

```
 4595  016562  011637  001106                  MOV     (SP),$LPADR      ;;SAVE SCOPE LOOP ADDRESS
 4596  016566  011637  001110                  MOV     (SP),$LPERR      ;;SAVE ERROR LOOP ADDRESS
 4597  016572  005037  001162                  CLR     $ESCAPE          ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
 4598  016576  112737  000001  001115          MOVB    #1,$ERMAX        ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
 4599  016604  013777  001102  162330 $OVER:   MOV     $TSTNM,@DISPLAY  ;;DISPLAY TEST NUMBER
 4600  016612  013716  001106                  MOV     $LPADR,(SP)      ;;FUDGE RETURN ADDRESS
 4601  016616  000002                          RTI                      ;;FIXES PS
 4602  016620  003720                 $MXCNT: 2000.                    ;;MAX. NUMBER OF ITERATIONS
```

```
                        .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE

                        ;;*********************************************************************
                        ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
                        ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
                        ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
                        ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
                        ;*REPLACED WITH SPACES.
                        ;*CALL:
                        ;*      MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE STACK
                        ;*      TYPDS                   ;;GO TO THE ROUTINE

016622                  $TYPDS:
016622  010046                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
016624  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
016626  010246                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
016630  010346                  MOV     R3,-(SP)        ;;PUSH R3 ON STACK
016632  010546                  MOV     R5,-(SP)        ;;PUSH R5 ON STACK
016634  012746  020200          MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
016640  016605  000020          MOV     20(SP),R5       ;;GET THE INPUT NUMBER
016644  100004                  BPL     1$              ;;BR IF INPUT IS POS.
016646  005405                  NEG     R5              ;;MAKE THE BINARY NUMBER POS.
016650  112766  000055  000001  MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
016656  005000          1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
016660  012703  017036          MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
016664  112723  000040          MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
016670  005002          2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
016672  016001  017026          MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
016676  160105          3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
016700  002402                  BLT     4$              ;;BR IF DONE
016702  005202                  INC     R2              ;;INCREASE THE BCD DIGIT BY 1
016704  000774                  BR      3$
016706  060105          4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
016710  005702                  TST     R2              ;;CHECK IF BCD DIGIT=0
016712  001002                  BNE     5$              ;;FALL THROUGH IF 0
016714  105716                  TSTB    (SP)            ;;STILL DOING LEADING 0'S?
016716  100407                  BMI     7$              ;;BR IF YES
016720  106316          5$:     ASLB    (SP)            ;;MSD?
016722  103003                  BCC     6$              ;;BR IF NO
016724  116663  000001  177777  MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
016732  052702  000060          BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
016736  052702  000040          BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
016742  110223                  MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
016744  005720                  TST     (R0)+           ;;JUST INCREMENTING
016746  020027  000010          CMP     R0,#10          ;;CHECK THE TABLE INDEX
016752  002746                  BLT     2$              ;;GO DO THE NEXT DIGIT
016754  003002                  BGT     8$              ;;GO TO EXIT
016756  010502                  MOV     R5,R2           ;;GET THE LSD
016760  000764                  BR      6$              ;;GO CHANGE TO ASCII
016762  105726          8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
016764  100003                  BPL     9$              ;;BR IF NO
016766  116663  177777  177776  MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
016774  105013          9$:     CLRB    (R3)            ;;SET THE TERMINATOR
016776  012605                  MOV     (SP)+,R5        ;;POP STACK INTO R5
017000  012603                  MOV     (SP)+,R3        ;;POP STACK INTO R3
017002  012602                  MOV     (SP)+,R2        ;;POP STACK INTO R2
```

```
 4659  017004  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
 4660  017006  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
 4661  017010  104401  017036                TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
 4662  017014  016666  000002  000004        MOV     2(SP),4(SP)     ;;ADJUST THE STACK
 4663  017022  012616                        MOV     (SP)+,(SP)
 4664  017024  000002                        RTI                     ;;RETURN TO USER
 4665  017026  023420                $DTBL:  10000.
 4666  017030  001750                        1000.
 4667  017032  000144                        100.
 4668  017034  000012                        10.
 4669  017036  000004                $DBLK:  .BLKW   4
```

```
                                .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE

                                ;;*****************************************************************
                                ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
                                ;*OCTAL (ASCII) NUMBER AND TYPE IT.
                                ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
                                ;*CALL:
                                ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                ;*      TYPOS                   ;;CALL FOR TYPEOUT
                                ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
                                ;*      .BYTE   M               ;;M=1 OR 0
                                ;*                                      ;;1=TYPE LEADING ZEROS
                                ;*                                      ;;0=SUPPRESS LEADING ZEROS
                                ;*
                                ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
                                ;*$TYPOS OR $TYPOC
                                ;*CALL:
                                ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                ;*      TYPON                   ;;CALL FOR TYPEOUT
                                ;*
                                ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
                                ;*CALL:
                                ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                ;*      TYPOC                   ;;CALL FOR TYPEOUT

017046  017646  000000          $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
017052  116637  000001  017271          MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
017060  112637  017273                  MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
017064  062716  000002                  ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
017070  000406                          BR      $TYPON
017072  112737  000001  017271  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
017100  112737  000006  017273          MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
017106  112737  000005  017270  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
017114  010346                          MOV     R3,-(SP)        ;;SAVE R3
017116  010446                          MOV     R4,-(SP)        ;;SAVE R4
017120  010546                          MOV     R5,-(SP)        ;;SAVE R5
017122  113704  017273                  MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
017126  005404                          NEG     R4
017130  062704  000006                  ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
017134  110437  017272                  MOVB    R4,$OMODE       ;;SAVE IT FOR USE
017140  113704  017271                  MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
017144  016605  000012                  MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
017150  005003                          CLR     R3              ;;CLEAR THE OUTPUT WORD
017152  006105                  1$:     ROL     R5              ;;ROTATE MSB INTO "C"
017154  000404                          BR      3$              ;;GO DO MSB
017156  006105                  2$:     ROL     R5              ;;FORM THIS DIGIT
017160  006105                          ROL     R5
017162  006105                          ROL     R5
017164  010503                          MOV     R5,R3
017166  006103                  3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
017170  105337  017272                  DECB    $OMODE          ;;TYPE THIS DIGIT?
017174  100016                          BPL     7$              ;;BR IF NO
017176  042703  177770                  BIC     #177770,R3      ;;GET RID OF JUNK
017202  001002                          BNE     4$              ;;TEST FOR 0
017204  005704                          TST     R4              ;;SUPPRESS THIS 0?
017206  001403                          BEQ     5$              ;;BR IF YES
```

```
 4726  017210  005204                 4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
 4727  017212  052703  000060                 BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
 4728  017216  052703  000040         5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
 4729  017222  110337  017266                 MOVB    R3,8$           ;;SAVE FOR TYPING
 4730  017226  104401  017266                 TYPE    ,8$             ;;GO TYPE THIS DIGIT
 4731  017232  105337  017270         7$:     DECB    $OCNT           ;;COUNT BY 1
 4732  017236  003347                         BGT     2$              ;;BR IF MORE TO DO
 4733  017240  002402                         BLT     6$              ;;BR IF DONE
 4734  017242  005204                         INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
 4735  017244  000744                         BR      2$              ;;GO DO THE LAST DIGIT
 4736  017246  012605                 6$:     MOV     (SP)+,R5        ;;RESTORE R5
 4737  017250  012604                         MOV     (SP)+,R4        ;;RESTORE R4
 4738  017252  012603                         MOV     (SP)+,R3        ;;RESTORE R3
 4739  017254  016666  000002  000004        MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
 4740  017262  012616                         MOV     (SP)+,(SP)
 4741  017264  000002                         RTI                     ;;RETURN
 4742  017266     000                 8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
 4743  017267     000                         .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
 4744  017270     000                 $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
 4745  017271     000                 $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
 4746  017272  000000                 $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
```

```
                                        .SBTTL  TRAP DECODER

                                        ;;**********************************************************************
                                        ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
                                        ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
                                        ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
                                        ;*GO TO THAT ROUTINE.

017274  010046                  $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
017276  016600  000002                  MOV     2(SP),R0        ;;GET TRAP ADDRESS
017302  005740                          TST     -(R0)           ;;BACKUP BY 2
017304  111000                          MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
017306  006300                          ASL     R0              ;;POSITION FOR INDEXING
017310  016000  017330                  MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
017314  000200                          RTS     R0              ;;GO TO ROUTINE

                                        ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO

017316  011646                  $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
017320  016666  000004  000002          MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
017326  000002                          RTI                     ;;RESTORE THE PSW

                                        .SBTTL  TRAP TABLE

                                        ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
                                        ;*BY THE "TRAP" INSTRUCTION.

                                        ;       ROUTINE
                                        ;       -------
017330  017316                  $TRPAD: .WORD   $TRAP2``
017332  014636                          $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
017334  017072                          $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
017336  017046                          $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
017340  017106                          $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
017342  016622                          $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)

017344  015170                          $GTSWR  ;;CALL=GTSWR    TRAP+6(104406)  GET SOFT-SWR SETTING

017346  015120                          $CKSWR  ;;CALL=CKSWR    TRAP+7(104407)  TEST FOR CHANGE IN SOFT-SWR
017350  015402                          $RDCHR  ;;CALL=RDCHR    TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
017352  015522                          $RDLIN  ;;CALL=RDLIN    TRAP+11(104411) TTY TYPEIN STRING ROUTINE
        000001          .END
```

```
ABASE = 176500            588#    932     973
ACDW1 = 000000            932
ACDW2 = 000000            932
ACPUOP= 000000            932     947
ACTCH   012476           3269*   3277*   3366    3715#
ADDRES  014310           4030*   4067*   4080*   4107*   4117    4143#
ADDW0 = 000000            932
ADDW1 = 000000            932
ADDW10= 000000            932
ADDW11= 000000            932
ADDW12= 000000            932
ADDW13= 000000            932
ADDW14= 000000            932
ADDW15= 000000            932
ADDW2 = 000000            932
ADDW3 = 000000            932
ADDW4 = 000000            932
ADDW5 = 000000            932
ADDW6 = 000000            932
ADDW7 = 000000            932
ADDW8 = 000000            932
ADDW9 = 000000            932
ADEVCT= 000000            932     938
ADEVM = 000000            588#    932     974
AENV  = 000000            932     943
AENVM = 000000            932     944
AFATAL= 000000            932     935
AMADR1= 000000            932     960
AMADR2= 000000            932     964
AMADR3= 000000            932     967
AMADR4= 000000            932     970
AMAMS1= 000000            932     954
AMAMS2= 000000            932     962
AMAMS3= 000000            932     965
AMAMS4= 000000            932     968
AMSGAD= 000000            932     940
AMSGLG= 000000            932     941
AMSGTY= 000000            932     934
AMTYP1= 000000            932     955
AMTYP2= 000000            932     963
AMTYP3= 000000            932     966
AMTYP4= 000000            932     969
APASS = 000000            932     937
APRIOR= 000000            932
APTCSU= 000040           4240    4484#
APTENV= 000001           4233    4440    4482#   4516
APTSIZ= 000200           1040    4481#
APTSPO= 000100           4235    4442    4483#
ASWREG= 000000            932     945
ATESTN= 000000            932     936
AUNIT = 000000            932     939
AUSWR = 001031            588#    932     946
AVECT1= 000300            588#    932     971
AVECT2= 000000            932     972
BIT0  = 000001            706#   2326    2351    2422    2650    2699    2907    3273    3343    3409    3988    4138
BIT00 = 000001            696#    706     752     772     792     813
```

```
BIT01 = 000002   695#   705   771   812
BIT02 = 000004   694#   704   770   811
BIT03 = 000010   693#   703   769   810
BIT04 = 000020   692#   702   768   809
BIT05 = 000040   691#   701   767   808
BIT06 = 000100   690#   700   746   766   786   807
BIT07 = 000200   689#   699   745   765   785   806
BIT08 = 000400   688#   698  4569
BIT09 = 001000   687#   697  4527  4577
BIT1  = 000002   705#  2657  2706  4139
BIT10 = 002000   686#  4504
BIT11 = 004000   685#  4584
BIT12 = 010000   684#   760  2811
BIT13 = 020000   683#   759  4511
BIT14 = 040000   682#   758  4555
BIT15 = 100000   681#   757  3434
BIT2  = 000004   704#  2673  2713  4140
BIT3  = 000010   703#  2683  2722  4141
BIT4  = 000020   702#
BIT5  = 000040   701#
BIT6  = 000100   700#  2911  2916  3053  3070  3086  3102  3118  3134  3150  3166  3280  3285
                3529  3557  3577  3605  3625  3653  3673  3701
BIT7  = 000200   699#
BIT8  = 000400   698#  3437  3490
BIT9  = 001000   697#
BPTVEC= 000014   713#
BREAK = 000001   792#  1218  1231  1234  1248  1251  1265  1270  2623  2736  2742  2758
BRK   = 000010   821#  1206  2600
CHAN    014316  1071*  4073*  4092  4095*  4105  4112  4116*  4120*  4146#
CHCNT   010100  2901*  2949*  2951  2959*  2965  3196#
CHCTR   012474  3266*  3300*  3302  3310*  3316  3403*  3412  3464*  3468  3477*  3483  3714#
CHMASK  010076  2899*  2907  2947*  2998*  3000*  3003  3195#
CHMSK   012472  3264*  3273  3296*  3405*  3409  3466*  3713#
CHNUM   013630  3978*  3990  3992*  3995*  4005#
CHOTAB  011472  3224  3321  3708#  3732
CH1TAB  011672  3323  3709#  3733
CH2TAB  012072  3325  3710#  3734
CH3TAB  012272  3327  3711#  3735
CKSWR = 104407  4500  4526  4554  4786#
CLR   = 000000   730#  2201
CONSOL  013736  1464  1599  1663  1725  1787  2098  2244  2377  2608  2956  3007  3307  3473
                4027*  4032*  4035#  4084*
CR    = 000015   621#  4279  4289
CRLF  = 000200   622#  1064  2817  2824  3935  3949  3975  3985  4155  4250  4289
CYCLE   013752  1075  4067#  4088  4099  4103  4121
DATA    005772  2420*  2449  2522  2528  2537#  2551  2556*  2565*  2567
DDISP = 177570   628#   913  1028
DISPLA  001142   913#  1028*  1036*  4503*  4599*
DISPRE  000174   832#  1036
DLADD   001254   993#  1079*  1085  1087  1090  1093  1126  1158  2832  2903  3261  4163
DLADDR= 176500   726#   993   995   996   997   998
DLVEC   001256   994#  1082*  1974  2109  2223  2403  2834  2879  3237  4165
DSWR  = 177570   627#   912  1027
EMTVEC= 000030   716#  1013*  1014*
ERRCHK  006612  2620*  2650*  2657*  2673*  2699  2706  2713  2722  2803#
ERRCNT  005732  2418*  2451  2462  2513#  2524  2560*  2683*
```

```
ERROR = 100000          757#  1870  1945  2312  2465  2549
ERRORF  003324         1477*  1527*  1554  1584#
ERRVEC= 000004          709#  1025  1026*  1037*  4560  4561*  4563*  4566*
ERWRD   012500         3219*  3437*  3445*  3458  3485  3490  3498  3716#
EVENOD= 000004          820#  2665
EXITFL  003326         1479*  1565*  1572*  1576  1585#
FLAG    012724         3778*  3805*  3810  3833  3845#
FRERR = 020000          759#  2476  2654
GNS   = ****** U        831  1063  2817  2824  2831  2833  2835  3935  3949  3975  3985  3987  4155
                       4157  4160  4162  4164  4778  4779  4780  4781  4782  4784  4786  4787  4788
GTSWR = 104406         1058  4784#
HOLDSC  012725         3790*  3795*  3801  3846#
HOWLON= 000000         3773#  3776
HT    = 000011          619#  4248  4289
I       001270          999#  1135*  1156*  1159  1161
ILLMEM= 000004          724#  1130  1183  3903  3924
INTFLA  013750         1142*  1147  1970*  2011  2035  2051*  2060  2117*  2147  2170  2193*  2202  3911*
                       3913  4052*  4055#
INTRTA  010056         2865  2979  2994  2997  3192#
INTSRV  013742          845  1131  1982  2110  3904  4049#
IOTVEC= 000020          714#  1011*  1012*
LF    = 000012          620#  4283  4289
LOOP    001732         1074#  2849  4205
MASK    014314         2899  2998  3264  3405  4082*  4085  4109  4145#
MASK1   013634         3951*  3956*  3976  4007#  4128
MASK2   013636         3952*  3957*  3961*  3962*  3963*  3964*  4008#  4129
MASK3   013640         3953*  3958*  3966*  4009#  4130
MASK4   013642         3954*  3959*  3968*  3969*  3970*  3971*  3972*  4010#  4131
MODNUM  013626         3977*  3985  3997*  3998  4004#
MODTST  007112         1107  2851#
MSK     013650         3980*  3988  3993*  4014#
MYTYPE  014324         4153#  4513
NUMBER  005734         2425*  2430*  2449  2514#  2522  2556  2567
ORERR = 040000          758#  1853  1890  1931  2468
PARITY= 000002          819#  2661
PASS    003322         1472*  1557*  1559  1583#
PERR  = 010000          760#  2484  2669  2680
PHASE2  014320         1069*  1105  2814  2841  2844*  4072*  4090  4101*  4147#
PIRQ  = 177772          626#
PIRQVE= 000240          720#
PRIERR  010102         2992*  3018*  3022  3197#
PR0   = 000000          643#  2003  2124  2401  2983  3369
PR1   = 000040          644#
PR2   = 000100          645#
PR3   = 000140          646#
PR4   = 000200          647#
PR5   = 000240          648#
PR6   = 000300          649#
PR7   = 000340          650#   846  1132  1287  1983  2111  2217  2408  2414  2876  2887  3243  3249
                       3257  3905
PS    = 177776          623#   624
PSW   = 177776          624#
PWRVEC= 000024          715#
P1CNT   014322         1072*  2821  2825*  4075*  4102*  4148#
RBUF    001262          996#  1087*  1088*  1168  1767  1849  1890  1931  1945  2309  2546  2644  2693
                       2768  2791
```

```
CVD
CVD

$MT
$MT
$MT
$MX
$NE

$NS

$NS

$NS
$NU
$NW

$OC
$OM
$OV
```

```
RCH0     010036            3045    3178#
RCH1     010040            3078    3179#
RCH2     010042            3110    3180#
RCH3     010044            3142    3181#
RCSR     001260             995#   1085*   1164    1367    1380*   1383    1397*   1400    1414*   1419    1620    1644    1684
                           1704*   1706    1746    1770    1822    1915    2120*   2141*   2190*   2292    2442*   2460*   2558*
                           2570*   2631    2776    3048    3065    3081    3097    3113    3129    3145    3161    3516    3544
                           3564    3592    3612    3640    3660    3688
RCVRDO=  000200             745#   1621    1644    1685    1706    1747    1770    1823    1916    2135    2293    2632    2777
                           2928
RCVRIE=  000100             746#   1367    1380    1383    1397    1400    1414    1419    2120    2141    2190    2442    2460
                           2558    2570
RCVROS   007536            2881    3042#
RCVR1S   007616            3075#
RCVR2S   007676            3107#
RCVR3S   007756            3139#
RDATA0=  000001             772#
RDATA1=  000002             771#
RDATA2=  000004             770#
RDATA3=  000010             769#
RDATA4=  000020             768#
RDATA5=  000040             767#
RDATA6=  000100             766#
RDATA7=  000200             765#
RDCHR  = 104410            4402    4787#
RDLIN  = 104411            4788#
RDRRUN=  000001             752#   1704
REC      005774            2406    2543#
REG    = 000004            3773#   3774
REGSAV   012720            3774*   3787    3843#
RESVEC=  000010             710#
RHLD     006100            2465    2468    2476    2484    2546*   2549    2551    2584#
ROSRV    010772            3235    3514#
R1SRV    011112            3562#
R2SRV    011232            3610#
R3SRV    011352            3658#
R5STAC   001332            1001#   1096
SEREND   011472            3253    3704#
SET    = 177777             729#   1493    1527    1538    1554    1565    1572    1576    1619    1683    1745    1821    1914
                           1990    2010    2059    2133    2146    2269    2291    2630    2740    2775    2926    3795
SETCLR=  000006            3773#   3801
SFTREG   013644            3909*   3915    3918*   4012#
SIZE     013012            1067    3898#
SRVEND   010036            2891    3169#
STACK  = 001100             614#   1009    1074
START    001334             835     850    1002#
STATEN   012472            3230    3712#
STKLMT=  177774             625#
SWR      001140             912#   1007    1027*   1029    1035*   1042*   1056    2811    4299    4336*   4504    4511    4523
                           4527    4555    4569    4571    4577    4584
SWREG    000176             833#   1035    1056    4299    4312
SW0    = 000001             678#
SW00   = 000001             668#    678
SW01   = 000002             667#    677
SW02   = 000004             666#    676
SW03   = 000010             665#    675
```

```
SW04   = 000020            664#    674
SW05   = 000040            663#    673
SW06   = 000100            662#    672
SW07   = 000200            661#    671
SW08   = 000400            660#    670
SW09   = 001000            659#    669
SW1    = 000002            677#
SW10   = 002000            658#
SW11   = 004000            657#
SW12   = 010000            656#
SW13   = 020000            655#
SW14   = 040000            654#
SW15   = 100000            653#
SW2    = 000004            676#
SW3    = 000010            675#
SW4    = 000020            674#
SW5    = 000040            673#
SW6    = 000100            672#
SW7    = 000200            671#
SW8    = 000400            670#
SW9    = 001000            669#
TABEND   010076           2871    3193#
TABL1    013652           3944    4016#
TABL2    014250           4080    4081    4123#
TABL3    014260           4082    4128#
TABL4    014270           4107    4108    4133#
TABL5    014300           4109    4138#
TABL6    012530           3418    3732#
TABL7    012540           3346    3348    3737#
TABL8    012550           3353    3355    3742#
TBITVE=  000014            711#
TBUF     001266            998#   1093*   1094*   1176    1487*   1509*   1518*   1612*   1677*   1739*   1817*   1839*   1911*
                          2055*   2127*   2196*   2287*   2528*   2626*   2771*
TCH0     010046           3062    3182#
TCH1     010050           3094    3183#
TCH2     010052           3126    3184#
TCH3     010054           3158    3185#
TCNT     010104           2989*   3002*   3009*   3010*   3012*   3014    3198#
TCSR     001264            997#   1090*   1091*   1172    1218    1231*   1234    1248*   1251    1265*   1270    1293    1306*
                          1309    1323*   1326    1340*   1345    1439    1494    1510    1523    1539    1991    1999*   2006*
                          2048*   2134    2270    2439*   2458*   2533*   2576*   2623*   2736*   2741    2758*
TDATA0=  000001            813#
TDATA1=  000002            812#
TDATA2=  000004            811#
TDATA3=  000010            810#
TDATA4=  000020            809#
TDATA5=  000040            808#
TDATA6=  000100            807#
TDATA7=  000200            806#
TEMP     010106           3048*   3050*   3053*   3065*   3067*   3070*   3081*   3083*   3086*   3097*   3099*   3102*   3113*
                          3115*   3118*   3129*   3131*   3134*   3145*   3147*   3150*   3161*   3163*   3166*   3199#
TIME   = 000000           3864#   3866
TIMER    012560           1497    1542    1623    1687    1749    1825    1918    1994    2014    2063    2137    2150    2205
                          2273    2295    2634    2744    2779    2930    3772#
TIMSAV   012722           3776*   3812    3823*   3844#
TKVEC  = 000060            718#   4031
```

```
TMP     012502          3516* 3518* 3520  3526* 3529* 3544* 3546* 3548* 3553* 3557* 3564* 3566* 3568
                        3574* 3577* 3592* 3594* 3596* 3601* 3605* 3612* 3614* 3616  3622* 3625* 3640*
                        3642* 3644* 3649* 3653* 3660* 3662* 3664  3670* 3673* 3688* 3690* 3692* 3697*
                        3701* 3719#
TMP0    012504          3333* 3394* 3424  3429  3522  3548  3550* 3721#
TMP0E   012506          3339  3396  3522  3722#
TMP1    012510          3396* 3570  3596  3598* 3723#
TMP1E   012512          3398  3570  3724#
TMP2    012514          3398* 3618  3644  3646* 3725#
TMP2E   012516          3400  3618  3726#
TMP3    012520          3400* 3666  3692  3694* 3727#
TMP3E   012522          3666  3728#
TMP4    012526          3366* 3373  3378* 3382* 3384  3535* 3583* 3631* 3679* 3730#
TMP6    013646          3908* 3912  3917* 4013#
TMP7    013740          4023* 4024* 4025  4036#
TPVEC = 000004           719#
TRAN    005736          2412  2520#
TRAPVE= 000034           717#  1015* 1016*
TRTVEC= 000014           712#
TRUE  = 000001           731#  1105  1464  1599  1663  1725  2098  2244  2377  2608  2814  2841  3473
                        3805  3810  3833  4032  4090  4101
TSTCON  013662          2954  3006  3305  3471  4023#  4114
TST1    002066          1122#
TST10   003460          1603  1656#
TST11   003606          1667  1718#
TST12   003732          1729  1782#
TST13   004242          1797  1804  1861  1880  1898  1940  1955  1965#
TST14   004570          2092#
TST15   005156          2101  2238#
TST16   005432          2249  2281  2303  2335  2345  2360  2371#
TST17   006102          2380  2510  2594#
TST2    002304          1201#
TST20   006614          2603  2611  2802  2808#  3507
TST21   007112          2858#
TST22   010110          3035  3214#
TST3    002460          1209  1282#
TST4    002634          1359#
TST5    002776          1431#
TST6    003036          1459#
TST7    003330          1467  1506  1582  1592#
TYPDS = 104405          2836  4194  4782#
TYPE  = 104401          1050  2817  2824  2831  2833  2835  2836  3935  3949  3975  3985  3987  3990
                        4000  4155  4157  4160  4162  4164  4165  4192  4195  4253  4310  4311  4314
                        4327  4338  4357  4406  4409  4413  4506  4514  4661  4730  4778#
TYPOC = 104402          2832  2834  4156  4161  4163  4165  4313  4779#
TYPON = 104404          4781#
TYPOS = 104403          4159  4780#
VECTOR  014312          4031* 4068* 4081* 4108* 4144#
WAIT    012732          1844  2030  2165  2762  3820  3863#
WHICHB= 000002          3773# 3787
WRAP  = 000020           822#  1597  1661  1723  1790  2096  2242  2375  2598  4096
XMITIE= 000100           786#  1293  1306  1309  1323  1326  1340  1345  1999  2006  2048  2439  2458
                        2533  2576
XMITRD= 000200           785#  1439  1495  1523  1540  1992  2271
XMITOS  007566          3059#
XMIT1S  007646          3091#
```

```
XMIT2S  007726            3123#
XMIT3S  010006            3155#
XOSRV   011046            3542#
X1SRV   011166            3590#
X2SRV   011306            3638#
X3SRV   011426            3686#
$APTHD  001000             872     878#
$ASTAT= ****** U          4462    4477
$ATYC   015720            4433    4435#
$ATY1   015674            4431#
$ATY3   015702            4238    4432#
$ATY4   015712            4434#   4519
$AUTOB  001134             909#   1060*   4307    4428
$BASE   001250             973#   3908    4067
$BDADR  001122             904#
$BDDAT  001126             906#
$BELL   001164             924#   4506    4539
$BGNLE= 177777             588#
$CHARC  015114            4255*   4265*   4272    4281*   4286#
$CKSWR  015120            4299#   4786
$CMTAG  001100             892#   1004    1005    1013    1017    1018    1019
$CM3  = 000000             922#
$CNTLG  015645            4310    4423#
$CNTLU  015640            4327    4422#
$CPUOP  001222             947#
$CRLF   001171             926#   2836    4000    4165    4254    4289    4338    4422    4514    4539
$DBLK   017036            4627    4661    4669#
$DEVCT  001204             938#   1103*
$DEVM   001252             974#   3898    3915*   3932    3944*   3951    3952    3953    3954
$DOAGN  014612            4188    4197    4203#
$DTBL   017026            4630    4665#
$ENDAD  014602             858    1048    4199#   4534
$ENDCT  014550            1017    4190#
$ENDMG  014621            4192    4207#
$ENULL  014616            4195    4206#
$ENV    001214             943#   1054    4233    4440    4464    4516
$ENVM   001215             944#   1040    4235    4240    4442
$EOP    014514            3949    4076    4180#
$EOPCT  014542            1017*   4187#   4191
$ERFLG  001103             895#   4501*   4539    4544    4573    4575    4581*   4603
$ERMAX  001115             901#   1020*   4575    4598*   4603
$ERRFL= 000000            1079#   1080    1082#   1083    1085#   1086    1087#   1089    1090#   1092    1093#   1095    1096#
                          1097    1103#   1104    1112#   1113    1126#   1127    1129#   1130#   1131#   1132#   1133#   1134
                          1135#   1136    1139#   1140    1142#   1143    1156#   1157    1158#   1160    1181#   1182#   1183#
                          1184#   1186#   1187#   1188#   1189#   1190    1215#   1216    1229#   1230    1231#   1232    1245#
                          1246    1248#   1249    1262#   1263    1265#   1266    1290#   1291    1304#   1305    1306#   1307
                          1320#   1321    1323#   1324    1337#   1338    1340#   1341    1364#   1365    1378#   1379    1380#
                          1381    1394#   1395    1397#   1398    1411#   1412    1414#   1415    1436#   1437    1472#   1473
                          1477#   1478    1479#   1480    1487#   1488    1492#   1518#   1519    1527#   1528    1537#   1557#
                          1558    1565#   1566    1572#   1573    1608#   1609    1612#   1613    1618#   1638#   1639    1672#
                          1673    1677#   1678    1682#   1704#   1705    1734#   1735    1739#   1740    1744#   1767#   1768
                          1811#   1812    1817#   1818    1820#   1839#   1840    1842#   1849#   1850    1868#   1869    1886#
                          1887    1904#   1905    1911#   1912    1913#   1970#   1971    1974#   1975    1977#   1978    1980#
                          1981#   1982#   1983#   1984#   1985    1986#   1987    1989#   1999#   2000    2006#   2007    2009#
                          2028#   2045#   2046    2048#   2049    2051#   2052    2055#   2056    2058#   2075#   2076#   2077#
                          2078#   2080#   2081#   2082#   2083#   2084    2108#   2109#   2110#   2111#   2112#   2113    2115#
```

```
                    2116    2117#   2118    2120#   2121    2127#   2128    2132#   2141#   2142    2145#   2163#   2186#
                    2187    2190#   2191    2193#   2194    2196#   2197    2200#   2223#   2224    2225#   2226#   2227#
                    2228#   2230#   2231#   2232#   2233#   2234    2256#   2257    2259    2263    2268#   2287#   2288
                    2290#   2309#   2310    2403#   2404    2406#   2407    2408#   2409    2412#   2413    2414#   2415
                    2418#   2419    2420#   2421    2425#   2426    2430#   2431    2439#   2440    2442#   2443    2458#
                    2459    2460#   2461    2528#   2529    2533#   2534    2546#   2547    2556#   2557    2558#   2559
                    2560#   2561    2565#   2566    2570#   2571    2576#   2577    2617#   2618    2620#   2621    2623#
                    2624    2626#   2627    2629#   2644#   2645    2650#   2651    2657#   2658    2673#   2674    2683#
                    2684    2733#   2734    2736#   2737    2739#   2758#   2759    2760#   2768#   2769    2771#   2772
                    2774#   2825#   2826    2844#   2845    2846#   2847    2865#   2866    2869#   2870    2879#   2880
                    2881#   2882    2885#   2886    2887#   2888    2889#   2890    2899#   2900    2901#   2902    2903#
                    2904    2911#   2912    2913#   2914    2916#   2917    2919#   2920    2922#   2923    2925#   2944#
                    2945    2947#   2948    2949#   2950    2954#   2959#   2960    2979#   2980    2989#   2990    2992#
                    2993    2994#   2995    2997#   2998    3045#   3046    3048#   3049    3050#   3051    3053#   3054
                    3062#   3063    3065#   3066    3067#   3068    3070#   3071    3078#   3079    3081#   3082    3083#
                    3084    3086#   3087    3094#   3095    3097#   3098    3099#   3100    3102#   3103    3110#   3111
                    3113#   3114    3115#   3116    3118#   3119    3126#   3127    3129#   3130    3131#   3132    3134#
                    3135    3142#   3143    3145#   3146    3147#   3148    3150#   3151    3158#   3159    3161#   3162
                    3163#   3164    3166#   3167    3219#   3220    3224#   3225    3228#   3229    3235#   3236    3237#
                    3238    3241#   3242    3243#   3244    3245#   3246    3247#   3248    3249#   3250    3251#   3252
                    3261#   3262    3264#   3265    3266#   3267    3269#   3270    3277#   3278    3280#   3281    3282#
                    3283    3285#   3286    3287#   3288    3292#   3293    3296#   3297    3300#   3301    3305#   3310#
                    3311    3321#   3322    3323#   3324    3325#   3326    3327#   3328    3333#   3334    3336#   3337
                    3339#   3340    3346#   3347    3348#   3349    3353#   3354    3355#   3356    3359#   3360    3366#
                    3367    3378#   3379    3382#   3383    3394#   3395    3396#   3397    3398#   3399    3400#   3401
                    3403#   3404    3405#   3406    3412#   3413    3415#   3416    3418#   3419    3421#   3422    3424#
                    3425    3426#   3427    3429#   3430    3437#   3438    3445#   3446    3450#   3451    3453#   3454
                    3464#   3465    3466#   3467    3471#   3477#   3478    3516#   3517    3518#   3519    3520#   3521
                    3526#   3527    3529#   3530    3535#   3536    3544#   3545    3546#   3547    3548#   3549    3550#
                    3551    3553#   3554    3557#   3558    3564#   3565    3566#   3567    3568#   3569    3574#   3575
                    3577#   3578    3583#   3584    3592#   3593    3594#   3595    3596#   3597    3598#   3599    3601#
                    3602    3605#   3606    3612#   3613    3614#   3615    3616#   3617    3622#   3623    3625#   3626
                    3631#   3632    3640#   3641    3642#   3643    3644#   3645    3646#   3647    3649#   3650    3653#
                    3654    3660#   3661    3662#   3663    3664#   3665    3670#   3671    3673#   3674    3679#   3680
                    3688#   3689    3690#   3691    3692#   3693    3694#   3695    3697#   3698    3701#   3702    3774#
                    3775    3776#   3777    3778#   3779    3790#   3791    3795#   3796    3805#   3806    3818#   3823#
                    3824    3836#   3837    3840#   3842    3866#   3867    3868#   3869    3871    3875    3876#   3877
                    3879    3883    3902#   3903#   3904#   3905#   3906#   3907    3922#   3923#   3924#   3925#   3927#
                    3928#   3929#   3930#   3931    4052#   4053
$ERROR  016142      1013    4499#
$ERRPC  001116       902#   4161    4508*   4509*   4510    4539
$ERRTB  001254       991#
$ERTTL  001112       899#   2836    2839*   4507*   4539
$ESCAP  001162       923#   1019*   4530    4532    4539    4597*
$ETABL  001214       942#
$ETEND  001254       884     975#
$FATAL  001176       935#   4157*   4159    4468*
$FFLG   016140      4431*   4434*   4462    4471*   4479#
$FILLC  001156       920#   4258    4289
$FILLS  001155       919#   4289
$F$AND= 000310       588#   1107    1149    1208    1220    1236    1253    1272    1295    1311    1328    1347    1369
                    1385    1402    1421    1441    1466    1525    1556    1561    1599    1646    1663    1708    1725
                    1772    1789    1792    1855    1872    1892    1933    1947    2037    2098    2172    2244    2314
                    2325    2328    2331    2353    2377    2424    2464    2467    2470    2478    2486    2524    2551
                    2569    2600    2610    2648    2656    2663    2667    2671    2682    2701    2708    2715    2724
                    2793    2813    2816    2823    2843    2909    2953    2958    3024    3275    3304    3309    3345
```

```
                     3411    3436    3444    3470    3475    3492    3500    3524    3572    3620    3668    3789    3803
                     3835
$FSBAD= 000401        588#   1107    1149    1208    1220    1236    1253    1272    1295    1311    1328    1347    1369
                     1385    1402    1421    1441    1466    1525    1556    1561    1602    1646    1666    1708    1728
                     1772    1789    1792    1855    1872    1892    1933    1947    2037    2101    2172    2247    2314
                     2325    2328    2331    2355    2380    2424    2464    2467    2470    2478    2486    2526    2554
                     2569    2603    2610    2648    2656    2663    2667    2671    2682    2701    2708    2715    2724
                     2793    2813    2816    2823    2843    2909    2953    2958    3024    3275    3304    3309    3345
                     3411    3436    3444    3470    3475    3492    3500    3524    3572    3620    3668    3789    3803
                     3835
$FSBLA= 000170         588#
$FSCAS= 000150         588#
$FSDEC= 000220         588#
$FSGOO= 000400         588#   1105    1147    1206    1218    1234    1251    1270    1293    1309    1326    1345    1367
                     1383    1400    1419    1439    1464    1502    1523    1545    1554    1559    1597    1628    1644
                     1661    1691    1706    1723    1753    1770    1787    1790    1830    1853    1870    1890    1923
                     1931    1945    2019    2035    2096    2155    2170    2209    2242    2276    2298    2312    2323
                     2326    2329    2351    2375    2422    2462    2465    2468    2476    2484    2522    2549    2567
                     2598    2608    2637    2646    2654    2661    2665    2669    2680    2699    2706    2713    2722
                     2749    2782    2791    2811    2814    2821    2841    2907    2934    2951    2956    3022    3273
                     3302    3307    3343    3409    3434    3442    3468    3473    3490    3498    3522    3570    3618
                     3666    3787    3801    3833
$FSIF = 000110         588#   1105    1109    1114    1147    1153    1206    1211    1218    1224    1234    1240    1251
                     1257    1270    1276    1293    1299    1309    1315    1326    1332    1345    1351    1367    1373
                     1383    1389    1400    1406    1419    1425    1439    1450    1464    1469    1502    1508    1523
                     1531    1545    1550    1552    1554    1559    1567    1569    1574    1597    1599    1605    1628
                     1633    1644    1650    1661    1663    1669    1691    1697    1706    1713    1723    1725    1731
                     1753    1759    1770    1777    1787    1790    1794    1799    1801    1806    1830    1835    1853
                     1863    1870    1882    1890    1900    1923    1928    1931    1942    1945    1952    2019    2024
                     2035    2041    2096    2098    2103    2155    2160    2170    2176    2209    2214    2242    2244
                     2251    2276    2283    2298    2305    2312    2318    2323    2326    2329    2337    2339    2347
                     2349    2351    2353    2362    2375    2377    2382    2422    2427    2432    2462    2465    2468
                     2474    2476    2482    2484    2490    2496    2500    2506    2508    2522    2524    2530    2549
                     2551    2562    2567    2572    2578    2580    2598    2600    2605    2608    2613    2615    2637
                     2642    2646    2652    2654    2659    2661    2665    2669    2675    2677    2680    2685    2687
                     2689    2699    2704    2706    2711    2713    2720    2722    2730    2749    2754    2782    2787
                     2791    2797    2811    2814    2818    2821    2827    2829    2838    2841    2848    2907    2934
                     2939    2941    2951    2956    2961    2963    3022    3032    3273    3289    3294    3302    3307
                     3312    3314    3343    3350    3357    3409    3434    3439    3442    3447    3462    3468    3473
                     3479    3481    3490    3496    3498    3504    3522    3531    3537    3570    3579    3585    3618
                     3627    3633    3666    3675    3681    3787    3792    3797    3801    3807    3833    3838
$FSINC= 000210         588#   2256    2365    3868    3876    3884    3887
$FSLOO= 000200         588#   1474    1576    1579    3784    3810    3825
$FSNAM= 000160         588#
$FSNO = 000403         588#   1147    1206    1218    1234    1251    1270    1293    1309    1326    1345    1367    1383
                     1400    1419    1439    1464    1523    1554    1559    1597    1599    1644    1661    1663    1706
                     1723    1725    1770    1787    1790    1853    1870    1890    1931    1945    2035    2096    2098
                     2170    2242    2244    2312    2323    2326    2329    2351    2353    2375    2377    2422    2462
                     2465    2468    2476    2484    2522    2524    2549    2551    2567    2598    2600    2608    2654
                     2661    2665    2669    2680    2699    2706    2713    2722    2811    2814    2821    2841    2907
                     2951    2956    3022    3273    3302    3307    3343    3409    3434    3468    3473    3490    3522
                     3570    3618    3666    3787
$FSOR - 000320         588#   1107    1149    1208    1220    1236    1253    1272    1295    1311    1328    1347    1369
                     1385    1402    1421    1441    1466    1525    1556    1561    1599    1646    1663    1708    1725
                     1772    1789    1792    1855    1872    1892    1933    1947    2037    2098    2172    2244    2314
                     2325    2328    2331    2353    2355    2377    2424    2464    2467    2470    2478    2486    2524
```

```
                    2526    2551    2569    2600    2610    2648    2656    2663    2667    2671    2682    2701    2708
                    2715    2724    2793    2813    2816    2823    2843    2909    2953    2958    3024    3275    3304
                    3309    3345    3411    3436    3444    3470    3475    3492    3500    3524    3572    3620    3668
                    3789    3803    3835
$F$RTN= 000300       588#    3773    3849    3864    3891    4154    4167
$F$SEL= 000140       588#
$F$UNT= 000130       588#    1137    1161    2447    2449    2867    2871    2883    2891    2905    2965    3226    3230
                    3239    3253    3271    3316    3341    3361    3371    3373    3380    3384    3407    3432    3456
                    3483
$F$WHI= 000120       588#    1599    1663    1725    2098    2244    2353    2377    2524    2551    2600
$F$YES= 000402       588#    1105    1147    1206    1218    1234    1251    1270    1293    1309    1326    1345    1367
                    1383    1400    1419    1439    1464    1523    1554    1559    1597    1599    1644    1661    1663
                    1706    1723    1725    1770    1787    1790    1853    1870    1890    1931    1945    2035    2096
                    2098    2170    2242    2244    2312    2323    2326    2329    2351    2353    2375    2377    2422
                    2462    2465    2468    2476    2484    2522    2524    2549    2551    2567    2598    2600    2608
                    2646    2654    2661    2665    2669    2680    2699    2706    2713    2722    2791    2811    2814
                    2821    2841    2907    2951    2956    3022    3273    3302    3307    3343    3409    3434    3442
                    3468    3473    3490    3498    3522    3570    3618    3666    3787    3801    3833
$GDADR  001120       903#
$GDDAT  001124       905#
$GET42  014572      4196#
$GTSWR  015170      4311#    4784
$HD   = 000000       599
$HIBTS  001000       879#
$ICNT   001104       896#    4588*   4589    4591*   4602
$IFLEV= 177777       588#    1105#   1114#   1147#   1153#   1206#   1211#   1218#   1224#   1234#   1240#   1251#   1257#
                    1270#   1276#   1293#   1299#   1309#   1315#   1326#   1332#   1345#   1351#   1367#   1373#   1383#
                    1389#   1400#   1406#   1419#   1425#   1439#   1450#   1464#   1469#   1502#   1508#   1523#   1545#
                    1550#   1552#   1554#   1559#   1567#   1574#   1597#   1605#   1628#   1633#   1644#   1650#   1661#
                    1669#   1691#   1697#   1706#   1713#   1723#   1731#   1753#   1759#   1770#   1777#   1787#   1790#
                    1799#   1806#   1830#   1835#   1853#   1863#   1870#   1882#   1890#   1900#   1923#   1928#   1931#
                    1942#   1945#   1952#   2019#   2024#   2035#   2041#   2096#   2103#   2155#   2160#   2170#   2176#
                    2209#   2214#   2242#   2251#   2276#   2283#   2298#   2305#   2312#   2318#   2323#   2326#   2329#
                    2337#   2347#   2355    2362#   2375#   2382#   2422#   2432#   2462#   2465#   2468#   2478    2486
                    2496#   2506#   2508#   2522#   2530#   2549#   2567#   2578#   2580#   2598#   2608#   2613#   2615#
                    2637#   2642#   2646#   2652#   2654#   2659#   2661#   2665#   2669#   2675#   2680#   2685#   2687#
                    2689#   2699#   2704#   2706#   2711#   2713#   2720#   2722#   2730#   2749#   2754#   2782#   2787#
                    2791#   2797#   2811#   2814#   2821#   2827#   2829#   2838#   2841#   2848#   2907#   2934#   2939#
                    2941#   2951#   2956#   2961#   2963#   3022#   3032#   3273#   3294#   3302#   3307#   3312#   3314#
                    3343#   3357#   3409#   3434#   3439#   3442#   3447#   3462#   3468#   3473#   3479#   3481#   3490#
                    3496#   3498#   3504#   3522#   3537#   3570#   3585#   3618#   3633#   3666#   3681#   3787#   3797#
                    3801#   3807#   3833#   3838#
$INTAG  001135       910#    4339    4428
$ISKO = 000001      1105#   1114    1147#   1153    1206#   1211    1218#   1224    1234#   1240    1251#   1257    1270#
                    1276    1293#   1299    1309#   1315    1326#   1332    1345#   1351    1367#   1373    1383#   1389
                    1400#   1406    1419#   1425    1439#   1450    1464#   1469    1502#   1508    1523#   1552    1554#
                    1574    1597#   1605    1628#   1633    1644#   1650    1661#   1669    1691#   1697    1706#   1713
                    1723#   1731    1753#   1759    1770#   1777    1787#   1806    1830#   1835    1853#   1863    1870#
                    1882    1890#   1900    1923#   1928    1931#   1942    1945#   1952    2019#   2024    2035#   2041
                    2096#   2103    2155#   2160    2170#   2176    2209#   2214    2242#   2251    2276#   2283    2298#
                    2305    2312#   2318    2323#   2355#   2362    2375#   2382    2422#   2432    2462#   2508    2522#
                    2530    2549#   2580    2598#   2615    2637#   2642    2646#   2652    2654#   2659    2661#   2689
                    2699#   2704    2706#   2711    2713#   2720    2722#   2730    2749#   2754    2782#   2787    2791#
                    2797    2811#   2838    2841#   2848    2907#   2941    2951#   2963    3022#   3032    3273#   3294
                    3302#   3314    3343#   3357    3409#   3462    3468#   3481    3490#   3496    3498#   3504    3522#
                    3537    3570#   3585    3618#   3633    3666#   3681    3787#   3797    3801#   3807    3833#   3838
```

$ISK1 = 000001
$ISK2 = 000001
$ITEMB 001114
$LF 001172
$LFLG 016137
$LOCTA= 177777

1545#	1550	1559#	1567	1790#	1799	2326#	2347	2465#	2506	2567#	2578	2608#	
2613	2665#	2687	2814#	2829	2934#	2939	2956#	2961	3307#	3312	3434#	3439	
3442#	3447	3473#	3479										
2329#	2337	2468#	2478#	2486#	2496	2669#	2675	2680#	2685	2821#	2827		
900#	4157	4510*	4518	4539									
927#	4289	4413	4422	4539									
4472*	4478#												
588#	1106	1107	1109	1110	1111	1114	1115	1137	1138	1148	1149	1153	
1154	1162	1163	1207	1208	1211	1212	1219	1220	1224	1225	1235	1236	
1240	1241	1252	1253	1257	1258	1271	1272	1276	1277	1294	1295	1299	
1300	1310	1311	1315	1316	1327	1328	1332	1333	1346	1347	1351	1352	
1368	1369	1373	1374	1384	1385	1389	1390	1401	1402	1406	1407	1420	
1421	1425	1426	1440	1441	1450	1451	1465	1466	1469	1470	1474	1475	
1502	1503	1508	1509	1524	1525	1531	1532	1533	1545	1546	1550	1551	
1552	1553	1555	1556	1560	1561	1567	1568	1569	1570	1571	1574	1575	
1577	1578	1579	1580	1581	1598	1599	1600	1601	1602	1605	1606	1628	
1629	1633	1634	1645	1646	1650	1651	1662	1663	1664	1665	1666	1669	
1670	1691	1692	1697	1698	1707	1708	1713	1714	1724	1725	1726	1727	
1728	1731	1732	1753	1754	1759	1760	1771	1772	1777	1778	1788	1789	
1791	1792	1794	1795	1796	1799	1800	1801	1802	1803	1806	1807	1830	
1831	1835	1836	1854	1855	1863	1864	1871	1872	1882	1883	1891	1892	
1900	1901	1923	1924	1928	1929	1932	1933	1942	1943	1946	1947	1952	
1953	2019	2020	2024	2025	2036	2037	2041	2042	2097	2098	2099	2100	
2101	2103	2104	2155	2156	2160	2161	2171	2172	2176	2177	2209	2210	
2214	2215	2243	2244	2245	2246	2247	2251	2252	2257	2258	2259	2260	
2261	2262	2263	2276	2277	2283	2284	2298	2299	2305	2306	2313	2314	
2318	2319	2324	2325	2327	2328	2330	2331	2337	2338	2339	2340	2341	
2347	2348	2349	2350	2351	2352	2353	2354	2355	2362	2363	2364	2365	
2366	2367	2376	2377	2378	2379	2380	2382	2383	2423	2424	2427	2428	
2429	2432	2433	2447	2448	2450	2451	2452	2453	2454	2463	2464	2466	
2467	2469	2470	2474	2475	2476	2477	2478	2482	2483	2484	2485	2486	
2490	2491	2492	2496	2497	2498	2499	2500	2501	2502	2506	2507	2508	
2509	2523	2524	2525	2526	2530	2531	2550	2551	2552	2553	2554	2562	
2563	2564	2568	2569	2572	2573	2574	2578	2579	2580	2581	2599	2600	
2601	2602	2603	2605	2606	2607	2609	2610	2613	2614	2615	2616	2637	
2638	2642	2643	2647	2648	2652	2653	2655	2656	2659	2660	2662	2663	
2666	2667	2670	2671	2675	2676	2677	2678	2679	2681	2682	2685	2686	
2687	2688	2689	2690	2700	2701	2704	2705	2707	2708	2711	2712	2714	
2715	2720	2721	2723	2724	2730	2731	2749	2750	2754	2755	2782	2783	
2787	2788	2792	2793	2797	2798	2812	2813	2815	2816	2818	2819	2820	
2822	2823	2827	2828	2829	2830	2838	2839	2842	2843	2848	2849	2867	
2868	2872	2873	2883	2884	2892	2893	2905	2906	2908	2909	2934	2935	
2939	2940	2941	2942	2952	2953	2957	2958	2961	2962	2963	2964	2966	
2967	3023	3024	3032	3033	3226	3227	3231	3232	3239	3240	3254	3255	
3271	3272	3274	3275	3289	3290	3291	3294	3295	3303	3304	3308	3309	
3312	3313	3314	3315	3317	3318	3341	3342	3344	3345	3350	3351	3352	
3357	3358	3362	3363	3371	3372	3374	3375	3380	3381	3385	3386	3407	
3408	3410	3411	3432	3433	3435	3436	3439	3440	3443	3444	3447	3448	
3457	3458	3459	3460	3461	3462	3463	3469	3470	3474	3475	3479	3480	
3481	3482	3484	3485	3486	3487	3488	3491	3492	3496	3497	3499	3500	
3504	3505	3523	3524	3531	3532	3533	3537	3538	3571	3572	3579	3580	
3581	3585	3586	3619	3620	3627	3628	3629	3633	3634	3667	3668	3675	
3676	3677	3681	3682	3772	3784	3785	3788	3789	3792	3793	3794	3797	
3798	3802	3803	3807	3808	3811	3812	3813	3814	3825	3826	3827	3834	
3835	3836	3837	3838	3839	3841	3842	3849	3850	3851	3852	3863	3869	
3870	3871	3872	3873	3874	3875	3877	3878	3879	3880	3881	3882	3883	

```
                    3884    3885    3886    3887    3888    3889    3891    3892    3893    4153    4167    4168    4169
$LPADR  001106       897#   1021*   4579*   4595*   4600    4602
$LPERR  001110       898#   1022*   1139*   1215*   1229*   1245*   1262*   1290*   1304*   1320*   1337*   1364*   1378*
                    1394*   1411*   1436*   1608*   1638*   1672*   1734*   1811*   1868*   1886*   1904*   1986*   2045*
                    2115*   2186*   2617*   2733*   4529    4579    4596*   4602
$LSTCN= 177777       588#   1105    1107    1110    1111    1114    1137    1138    1147    1149    1153    1161    1206
                    1208    1211    1218    1220    1224    1234    1236    1240    1251    1253    1257    1270    1272
                    1276    1293    1295    1299    1309    1311    1315    1326    1328    1332    1345    1347    1351
                    1367    1369    1373    1383    1385    1389    1400    1402    1406    1419    1421    1425    1439
                    1441    1450    1464    1466    1469    1474    1475    1502    1503    1508    1523    1525    1532
                    1533    1545    1546    1550    1552    1554    1556    1559    1561    1567    1570    1571    1574
                    1579    1580    1597    1602    1605    1628    1629    1633    1644    1646    1650    1661    1666
                    1669    1691    1692    1697    1706    1708    1713    1723    1728    1731    1753    1754    1759
                    1770    1772    1777    1787    1789    1790    1792    1795    1796    1799    1802    1803    1806
                    1830    1831    1835    1853    1855    1863    1870    1872    1882    1890    1892    1900    1923
                    1924    1928    1931    1933    1942    1945    1947    1952    2019    2020    2024    2035    2037
                    2041    2096    2101    2103    2155    2156    2160    2170    2172    2176    2209    2210    2214
                    2242    2247    2251    2256    2258    2259    2260    2263    2276    2277    2283    2298    2299
                    2305    2312    2314    2318    2323    2325    2326    2328    2329    2331    2337    2340    2341
                    2347    2350    2351    2355    2362    2363    2365    2366    2375    2380    2382    2422    2424
                    2428    2429    2432    2447    2448    2449    2454    2462    2464    2465    2467    2468    2470
                    2475    2476    2478    2483    2484    2486    2491    2492    2496    2497    2498    2501    2502
                    2506    2508    2522    2526    2530    2549    2554    2563    2564    2567    2569    2573    2574
                    2578    2580    2598    2603    2606    2607    2608    2610    2613    2615    2637    2638    2642
                    2646    2648    2652    2654    2656    2659    2661    2663    2665    2667    2669    2671    2675
                    2678    2679    2680    2682    2685    2687    2689    2699    2701    2704    2706    2708    2711
                    2713    2715    2720    2722    2724    2730    2749    2750    2754    2782    2783    2787    2791
                    2793    2797    2811    2813    2814    2816    2819    2820    2821    2823    2827    2829    2838
                    2841    2843    2848    2867    2868    2871    2883    2884    2891    2905    2906    2907    2909
                    2934    2935    2939    2941    2951    2953    2956    2958    2961    2963    2965    3022    3024
                    3032    3226    3227    3230    3239    3240    3253    3271    3272    3273    3275    3290    3291
                    3294    3302    3304    3307    3309    3312    3314    3316    3341    3342    3343    3345    3351
                    3352    3357    3361    3371    3372    3373    3380    3381    3384    3407    3408    3409    3411
                    3432    3433    3434    3436    3439    3442    3444    3447    3456    3461    3462    3468    3470
                    3473    3475    3479    3481    3483    3488    3490    3492    3496    3498    3500    3504    3522
                    3524    3532    3533    3537    3570    3572    3580    3581    3585    3618    3620    3628    3629
                    3633    3666    3668    3676    3677    3681    3773    3784    3785    3787    3789    3793    3794
                    3797    3801    3803    3807    3825    3826    3833    3835    3838    3849    3864    3868    3870
                    3871    3872    3875    3876    3878    3879    3880    3883    3884    3885    3887    3888    3891
                    4154    4167
$LSTIN- 000000       588#   1079    1080    1082    1083    1085    1086    1087    1088    1089    1090    1091    1092
                    1093    1094    1095    1096    1097    1103    1104    1105    1106    1107    1109    1110    1112
                    1113    1126    1127    1129    1130    1131    1132    1133    1134    1135    1136    1139    1140
                    1142    1143    1147    1148    1149    1156    1157    1158    1159    1160    1161    1162    1163
                    1181    1182    1183    1184    1185    1186    1187    1188    1189    1190    1206    1207    1208
                    1215    1216    1218    1219    1220    1229    1230    1231    1232    1234    1235    1236    1245
                    1246    1248    1249    1251    1252    1253    1262    1263    1265    1266    1270    1271    1272
                    1290    1291    1293    1294    1295    1304    1305    1306    1307    1309    1310    1311    1320
                    1321    1323    1324    1326    1327    1328    1337    1338    1340    1341    1345    1346    1347
                    1364    1365    1367    1368    1369    1378    1379    1380    1381    1383    1384    1385    1394
                    1395    1397    1398    1400    1401    1402    1411    1412    1414    1415    1419    1420    1421
                    1436    1437    1439    1440    1441    1464    1465    1466    1472    1473    1477    1478    1479
                    1480    1487    1488    1492    1493    1494    1495    1496    1497    1498    1499    1502    1503
                    1518    1519    1523    1524    1525    1527    1528    1531    1532    1537    1538    1539    1540
                    1541    1542    1543    1544    1545    1546    1554    1555    1556    1557    1558    1559    1560
                    1561    1565    1566    1569    1570    1572    1573    1576    1577    1578    1579    1580    1597
```

1598	1599	1600	1601	1608	1609	1612	1613	1618	1619	1620	1621	1622
1623	1624	1625	1628	1629	1638	1639	1644	1645	1646	1661	1662	1663
1664	1665	1672	1673	1677	1678	1682	1683	1684	1685	1686	1687	1688
1689	1691	1692	1704	1705	1706	1707	1708	1723	1724	1725	1726	1727
1734	1735	1739	1740	1744	1745	1746	1747	1748	1749	1750	1751	1753
1754	1767	1768	1770	1771	1772	1787	1788	1789	1790	1791	1792	1794
1795	1801	1802	1811	1812	1817	1818	1820	1821	1822	1823	1824	1825
1826	1827	1830	1831	1839	1840	1842	1843	1844	1845	1846	1849	1850
1853	1854	1855	1868	1869	1870	1871	1872	1886	1887	1890	1891	1892
1904	1905	1911	1912	1913	1914	1915	1916	1917	1918	1919	1920	1923
1924	1931	1932	1933	1945	1946	1947	1970	1971	1974	1975	1977	1978
1980	1981	1982	1983	1984	1985	1986	1987	1989	1990	1991	1992	1993
1994	1995	1996	1999	2000	2006	2007	2009	2010	2011	2012	2013	2014
2015	2016	2019	2020	2028	2029	2030	2031	2032	2035	2036	2037	2045
2046	2048	2049	2051	2052	2055	2056	2058	2059	2060	2061	2062	2063
2064	2065	2075	2076	2077	2078	2079	2080	2081	2082	2083	2084	2096
2097	2098	2099	2100	2108	2109	2110	2111	2112	2113	2115	2116	2117
2118	2120	2121	2127	2128	2132	2133	2134	2135	2136	2137	2138	2139
2141	2142	2145	2146	2147	2148	2149	2150	2151	2152	2155	2156	2163
2164	2165	2166	2167	2170	2171	2172	2186	2187	2190	2191	2193	2194
2196	2197	2200	2201	2202	2203	2204	2205	2206	2207	2209	2210	2223
2224	2225	2226	2227	2228	2229	2230	2231	2232	2233	2234	2242	2243
2244	2245	2246	2256	2257	2258	2259	2260	2261	2262	2263	2268	2269
2270	2271	2272	2273	2274	2275	2276	2277	2287	2288	2290	2291	2292
2293	2294	2295	2296	2297	2298	2299	2309	2310	2312	2313	2314	2323
2324	2325	2326	2327	2328	2329	2330	2331	2339	2340	2349	2350	2351
2352	2353	2354	2355	2365	2366	2375	2376	2377	2378	2379	2403	2404
2406	2407	2408	2409	2412	2413	2414	2415	2418	2419	2420	2421	2422
2423	2424	2425	2426	2427	2428	2430	2431	2439	2440	2442	2443	2449
2450	2451	2452	2453	2458	2459	2460	2461	2462	2463	2464	2465	2466
2467	2468	2469	2470	2474	2475	2476	2477	2478	2482	2483	2484	2485
2486	2490	2491	2500	2501	2522	2523	2524	2525	2526	2528	2529	2533
2534	2546	2547	2549	2550	2551	2552	2553	2556	2557	2558	2559	2560
2561	2562	2563	2565	2566	2567	2568	2569	2570	2571	2572	2573	2576
2577	2598	2599	2600	2601	2602	2605	2606	2608	2609	2610	2617	2618
2620	2621	2623	2624	2626	2627	2629	2630	2631	2632	2633	2634	2635
2636	2637	2638	2644	2645	2646	2647	2648	2650	2651	2654	2655	2656
2657	2658	2661	2662	2663	2665	2666	2667	2669	2670	2671	2673	2674
2677	2678	2680	2681	2682	2683	2684	2699	2700	2701	2706	2707	2708
2713	2714	2715	2722	2723	2724	2733	2734	2736	2737	2739	2740	2741
2742	2743	2744	2745	2746	2749	2750	2758	2759	2760	2761	2762	2763
2764	2768	2769	2771	2772	2774	2775	2776	2777	2778	2779	2780	2781
2782	2783	2791	2792	2793	2811	2812	2813	2814	2815	2816	2818	2819
2821	2822	2823	2825	2826	2841	2842	2843	2844	2845	2846	2847	2865
2866	2869	2870	2871	2872	2873	2879	2880	2881	2882	2885	2886	2887
2888	2889	2890	2891	2892	2893	2899	2900	2901	2902	2903	2904	2907
2908	2909	2911	2912	2913	2914	2916	2917	2919	2920	2922	2923	2925
2926	2927	2928	2929	2930	2931	2932	2934	2935	2944	2945	2947	2948
2949	2950	2951	2952	2953	2954	2955	2956	2957	2958	2959	2960	2965
2966	2967	2979	2980	2989	2990	2992	2993	2994	2995	2997	2998	3022
3023	3024	3045	3046	3048	3049	3050	3051	3053	3054	3062	3063	3065
3066	3067	3068	3070	3071	3078	3079	3081	3082	3083	3084	3086	3087
3094	3095	3097	3098	3099	3100	3102	3103	3110	3111	3113	3114	3115
3116	3118	3119	3126	3127	3129	3130	3131	3132	3134	3135	3142	3143
3145	3146	3147	3148	3150	3151	3158	3159	3161	3162	3163	3164	3166
3167	3219	3220	3224	3225	3228	3229	3230	3231	3232	3235	3236	3237

	3238	3241	3242	3243	3244	3245	3246	3247	3248	3249	3250	3251	3252
	3253	3254	3255	3261	3262	3264	3265	3266	3267	3269	3270	3273	3274
	3275	3277	3278	3280	3281	3282	3283	3285	3286	3287	3288	3289	3290
	3292	3293	3296	3297	3300	3301	3302	3303	3304	3305	3306	3307	3308
	3309	3310	3311	3316	3317	3318	3321	3322	3323	3324	3325	3326	3327
	3328	3333	3334	3336	3337	3339	3340	3343	3344	3345	3346	3347	3348
	3349	3350	3351	3353	3354	3355	3356	3359	3360	3361	3362	3363	3366
	3367	3373	3374	3375	3378	3379	3382	3383	3384	3385	3386	3394	3395
	3396	3397	3398	3399	3400	3401	3403	3404	3405	3406	3409	3410	3411
	3412	3413	3415	3416	3418	3419	3421	3422	3424	3425	3426	3427	3429
	3430	3434	3435	3436	3437	3438	3442	3443	3444	3445	3446	3450	3451
	3453	3454	3456	3457	3458	3459	3460	3464	3465	3466	3467	3468	3469
	3470	3471	3472	3473	3474	3475	3477	3478	3483	3484	3485	3486	3487
	3490	3491	3492	3498	3499	3500	3516	3517	3518	3519	3520	3521	3522
	3523	3524	3526	3527	3529	3530	3531	3532	3535	3536	3544	3545	3546
	3547	3548	3549	3550	3551	3553	3554	3557	3558	3564	3565	3566	3567
	3568	3569	3570	3571	3572	3574	3575	3577	3578	3579	3580	3583	3584
	3592	3593	3594	3595	3596	3597	3598	3599	3601	3602	3605	3606	3612
	3613	3614	3615	3616	3617	3618	3619	3620	3622	3623	3625	3626	3627
	3628	3631	3632	3640	3641	3642	3643	3644	3645	3646	3647	3649	3650
	3653	3654	3660	3661	3662	3663	3664	3665	3666	3667	3668	3670	3671
	3673	3674	3675	3676	3679	3680	3688	3689	3690	3691	3692	3693	3694
	3695	3697	3698	3701	3702	3774	3775	3776	3777	3778	3779	3787	3788
	3789	3790	3791	3792	3793	3795	3796	3801	3802	3803	3805	3806	3810
	3811	3812	3813	3814	3818	3819	3820	3821	3822	3823	3824	3825	3826
	3833	3834	3835	3836	3837	3840	3841	3842	3850	3851	3852	3853	3866
	3867	3868	3869	3870	3871	3872	3873	3874	3875	3876	3877	3878	3879
	3880	3881	3882	3883	3884	3885	3887	3888	3893	3894	3902	3903	3904
	3905	3906	3907	3922	3923	3924	3925	3926	3927	3928	3929	3930	3931
	4052	4053	4169	4170									
$LSTST= 177777	588#	1105	1107	1109	1110	1111	1114	1137	1138	1147	1149	1153	1161
	1206	1208	1211	1218	1220	1224	1234	1236	1240	1251	1253	1257	1270
	1272	1276	1293	1295	1299	1309	1311	1315	1326	1328	1332	1345	1347
	1351	1367	1369	1373	1383	1385	1389	1400	1402	1406	1419	1421	1425
	1439	1441	1450	1464	1466	1469	1474	1475	1502	1503	1508	1523	1525
	1531	1532	1533	1545	1546	1550	1552	1554	1556	1559	1561	1567	1569
	1570	1571	1574	1576	1579	1580	1597	1602	1605	1628	1629	1633	1644
	1646	1650	1661	1666	1669	1691	1692	1697	1706	1708	1713	1723	1728
	1731	1753	1754	1759	1770	1772	1777	1787	1789	1790	1792	1794	1795
	1796	1799	1801	1802	1803	1806	1830	1831	1835	1853	1855	1863	1870
	1872	1882	1890	1892	1900	1923	1924	1928	1931	1933	1942	1945	1947
	1952	2019	2020	2024	2035	2037	2041	2096	2101	2103	2155	2156	2160
	2170	2172	2176	2209	2210	2214	2242	2247	2251	2256	2258	2259	2260
	2263	2276	2277	2283	2298	2299	2305	2312	2314	2318	2323	2325	2326
	2328	2329	2331	2337	2339	2340	2341	2347	2349	2350	2351	2355	2362
	2363	2365	2366	2375	2380	2382	2422	2424	2427	2428	2429	2432	2447
	2448	2449	2462	2464	2465	2467	2468	2470	2474	2475	2476	2478	2482
	2483	2484	2486	2490	2491	2492	2496	2497	2498	2500	2501	2502	2506
	2508	2522	2526	2530	2549	2554	2562	2563	2564	2567	2569	2572	2573
	2574	2578	2580	2598	2603	2605	2606	2607	2608	2610	2613	2615	2637
	2638	2642	2646	2648	2652	2654	2656	2659	2661	2663	2665	2667	2669
	2671	2675	2677	2678	2679	2680	2682	2685	2687	2689	2699	2701	2704
	2706	2708	2711	2713	2715	2720	2722	2724	2730	2749	2750	2754	2782
	2783	2787	2791	2793	2797	2811	2813	2814	2816	2818	2819	2820	2821
	2823	2827	2829	2838	2841	2843	2848	2867	2868	2871	2883	2884	2891
	2905	2906	2907	2909	2934	2935	2939	2941	2951	2953	2956	2958	2961

```
                    2963    2965    3022    3024    3032    3226    3227    3230    3239    3240    3253    3271    3272
                    3273    3275    3289    3290    3291    3294    3302    3304    3307    3309    3312    3314    3316
                    3341    3342    3343    3345    3350    3351    3352    3357    3361    3371    3372    3373    3380
                    3381    3384    3407    3408    3409    3411    3432    3433    3434    3436    3439    3442    3444
                    3447    3456    3462    3468    3470    3473    3475    3479    3481    3483    3490    3492    3496
                    3498    3500    3504    3522    3524    3531    3532    3533    3537    3570    3572    3579    3580
                    3581    3585    3618    3620    3627    3628    3629    3633    3666    3668    3675    3676    3677
                    3681    3773    3784    3785    3787    3789    3792    3793    3794    3797    3801    3803    3807
                    3810    3825    3826    3833    3835    3838    3849    3864    3868    3870    3871    3872    3875
                    3876    3878    3879    3880    3883    3884    3885    3887    3888    3891    4154    4167
$LSTTA= 000000       588#    1110    1111    1114    1115    1137    1138    1153    1154    1211    1212    1224    1225
                    1240    1241    1257    1258    1276    1277    1299    1300    1315    1316    1332    1333    1351
                    1352    1373    1374    1389    1390    1406    1407    1425    1426    1450    1451    1469    1470
                    1474    1475    1508    1509    1532    1533    1550    1551    1552    1553    1567    1568    1570
                    1571    1574    1575    1580    1581    1601    1602    1605    1606    1633    1634    1650    1651
                    1665    1666    1669    1670    1697    1698    1713    1714    1727    1728    1731    1732    1759
                    1760    1777    1778    1795    1796    1799    1800    1802    1803    1806    1807    1835    1836
                    1863    1864    1882    1883    1900    1901    1928    1929    1942    1943    1952    1953    2024
                    2025    2041    2042    2100    2101    2103    2104    2160    2161    2176    2177    2214    2215
                    2246    2247    2251    2252    2258    2259    2260    2261    2283    2284    2305    2306    2318
                    2319    2337    2338    2340    2341    2347    2348    2350    2351    2362    2363    2364    2366
                    2367    2379    2380    2382    2383    2428    2429    2432    2433    2447    2448    2453    2454
                    2475    2476    2483    2484    2491    2492    2496    2497    2498    2499    2501    2502    2506
                    2507    2508    2509    2530    2531    2553    2554    2563    2564    2573    2574    2578    2579
                    2580    2581    2602    2603    2606    2607    2613    2614    2615    2616    2642    2643    2652
                    2653    2659    2660    2675    2676    2678    2679    2685    2686    2687    2688    2689    2690
                    2704    2705    2711    2712    2720    2721    2730    2731    2754    2755    2787    2788    2797
                    2798    2819    2820    2827    2828    2829    2830    2838    2839    2848    2849    2867    2868
                    2883    2884    2905    2906    2939    2940    2941    2942    2961    2962    2963    2964    3032
                    3033    3226    3227    3239    3240    3271    3272    3290    3291    3294    3295    3312    3313
                    3314    3315    3341    3342    3351    3352    3357    3358    3371    3372    3380    3381    3407
                    3408    3432    3433    3439    3440    3447    3448    3460    3461    3462    3463    3479    3480
                    3481    3482    3487    3488    3496    3497    3504    3505    3532    3533    3537    3538    3580
                    3581    3585    3586    3628    3629    3633    3634    3676    3677    3681    3682    3772    3773
                    3784    3785    3793    3794    3797    3798    3807    3808    3826    3827    3838    3839    3849
                    3850    3851    3852    3863    3864    3870    3871    3872    3873    3878    3879    3880    3881
                    3885    3886    3888    3889    3891    3892    3893    4153    4154    4167    4168    4169
$MADR1  001226       960#
$MADR2  001232       964#
$MADR3  001236       967#
$MADR4  001242       970#
$MAIL   001174       880     884     933#    1039    1054    1124    1203    1284    1361    1433    1461    1594    1658
                    1720    1784    1967    2094    2240    2373    2596    2860    3216    4233    4516    4594
$MAMS1  001224       954#
$MAMS2  001230       962#
$MAMS3  001234       965#
$MAMS4  001240       968#
$MBADR  001002       880#
$MCALL= 000000         1#    588
$MFLG   016136      4432*   4438    4473*   4477#
$MNEW   015663      4314    4426#
$MSGAD  001210       940#   4448*   4451
$MSGLG  001212       941#   4453*
$MSGTY  001174       934#   4446    4454*   4466    4470*
$MSWR   015652      4311    4424#
$MTYP1  001225       955#
```

```
$MTYP2  001231            963#
$MTYP3  001235            966#
$MTYP4  001241            969#
$MXCNT  016620           4592    4602#
$NESTL= 177777            588#  1105#  1109    1114#  1137#  1147#  1153#  1161#  1206#  1211#  1218#  1224#  1234#
                         1240#  1251#  1257#  1270#  1276#  1293#  1299#  1309#  1315#  1326#  1332#  1345#  1351#
                         1367#  1373#  1383#  1389#  1400#  1406#  1419#  1425#  1439#  1450#  1464#  1469#  1474#
                         1502#  1508#  1523#  1531   1545#  1550#  1552#  1554#  1559#  1567#  1569   1574#  1576
                         1579#  1597#  1605#  1628#  1633#  1644#  1650#  1661#  1669#  1691#  1697#  1706#  1713#
                         1723#  1731#  1753#  1759#  1770#  1777#  1787#  1790#  1794   1799#  1801   1806#  1830#
                         1835#  1853#  1863#  1870#  1882#  1890#  1900#  1923#  1928#  1931#  1942#  1945#  1952#
                         2019#  2024#  2035#  2041#  2096#  2103#  2155#  2160#  2170#  2176#  2209#  2214#  2242#
                         2251#  2256#  2276#  2283#  2298#  2305#  2312#  2318#  2323#  2326#  2329#  2337#  2339
                         2347#  2349   2362#  2365#  2375#  2382#  2422#  2427   2432#  2447#  2449#  2462#  2465#
                         2468#  2474   2482   2490   2496#  2500   2506#  2508#  2522#  2530#  2549#  2562   2567#
                         2572   2578#  2580#  2598#  2605   2608#  2613#  2615#  2637#  2642#  2646#  2652#  2654#
                         2659#  2661#  2665#  2669#  2675#  2677   2680#  2685#  2687#  2689#  2699#  2704#  2706#
                         2711#  2713#  2720#  2722#  2730#  2749#  2754#  2782#  2787#  2791#  2797#  2811#  2814#
                         2818   2821#  2827#  2829#  2838#  2841#  2848#  2867#  2871#  2883#  2891#  2905#  2907#
                         2934#  2939#  2941#  2951#  2956#  2961#  2963#  2965#  3022#  3032#  3226#  3230#  3239#
                         3253#  3271#  3273#  3289   3294#  3302#  3307#  3312#  3314#  3316#  3341#  3343#  3350
                         3357#  3361#  3371#  3373#  3380#  3384#  3407#  3409#  3432#  3434#  3439#  3442#  3447#
                         3456#  3462#  3468#  3473#  3479#  3481#  3483#  3490#  3496#  3498#  3504#  3522#  3531
                         3537#  3570#  3579   3585#  3618#  3627   3633#  3666#  3675   3681#  3773#  3784#  3787#
                         3792   3797#  3801#  3807#  3810   3825#  3833#  3838#  3849#  3864#  3868#  3876#  3884#
                         3887#  3891#  4154#  4167#
$NSK0 = 000300           1105#  1109   1114   1137#  1161   1206#  1211   1218#  1224   1234#  1240   1251#  1257
                         1270#  1276   1293#  1299   1309#  1315   1326#  1332   1345#  1351   1367#  1373   1383#
                         1389   1400#  1406   1419#  1425   1439#  1450   1464#  1469   1474#  1576   1579   1597#
                         1605   1628#  1633   1644#  1650   1661#  1669   1691#  1697   1706#  1713   1723#  1731
                         1753#  1759   1770#  1777   1787#  1801   1806   1830#  1835   1853#  1863   1870#  1882
                         1890#  1900   1923#  1928   1931#  1942   1945#  1952   2019#  2024   2035#  2041   2096#
                         2103   2155#  2160   2170#  2176   2209#  2214   2242#  2251   2256#  2365   2375#  2382
                         2422#  2427   2432   2447#  2449   2462#  2508   2522#  2530   2549#  2562   2580   2598#
                         2605   2615   2637#  2642   2646#  2652   2654#  2659   2661#  2689   2699#  2704   2706#
                         2711   2713#  2720   2722#  2730   2749#  2754   2782#  2787   2791#  2797   2811#  2838
                         2841#  2848   2867#  2871   2883#  2891   2905#  2965   3022#  3032   3226#  3230   3239#
                         3253   3271#  3316   3341#  3361   3371#  3373   3380#  3384   3407#  3483   3490#  3496
                         3498#  3504   3522#  3531   3537   3570#  3579   3585   3618#  3627   3633   3666#  3675
                         3681   3773#  3849   3864#  3891   4154#  4167
$NSK1 = 000210           1147#  1153   1502#  1508   1523#  1531   1552   1554#  1569   1574   1790#  1794   1799
                         2276#  2283   2298#  2305   2312#  2318   2323#  2349   2362   2465#  2500   2506   2567#
                         2572   2578   2608#  2613   2665#  2677   2687   2814#  2818   2829   2907#  2941   2951#
                         2963   3273#  3289   3294   3302#  3314   3343#  3350   3357   3409#  3462   3468#  3481
                         3784#  3810   3825   3833#  3838   3868#  3887
$NSK2 = 000210           1545#  1550   1559#  1567   2326#  2339   2347   2468#  2474   2482   2490   2496   2669#
                         2675   2680#  2685   2821#  2827   2934#  2939   2956#  2961   3307#  3312   3432#  3456
                         3473#  3479   3787#  3792   3797   3801#  3807   3876#  3884
$NSK3 = 000110           2329#  2337   3434#  3439   3442#  3447
$NULL   001154            918#  4260   4289
$NWTST= 000001           1115#  1117   1196#  1198   1279#  1354#  1356   1428#  1454#  1456   1587#  1589   1653#
                         1715#  1779#  1957#  1959   2086#  2088   2235#  2368#  2587#  2589   2805#  2852#  2854
                         3202#  3204
$OCNT   017270           4702*  4731*  4744#
$OMODE  017272           4697*  4701*  4706   4709*  4720*  4746#
$OVER   016604           4556   4572   4580   4590   4599#
```

```
$PASS   001202            937#  1039*  4184*  4185*  4193   4206   4586   4603
$PASTM  001006            882#
$QUES   001170            925#  4289   4357   4406   4422   4539
$RDCHR  015402           4370#  4787
$RDDEC= ****** U         4789
$RDLIN  015522           4398#  4788
$RDOCT= ****** U         4789
$RDSZ = 000010           4391#
$RTNAD  014614           4205#
$R2A  = ****** U         4789
$SAVLE= 177777            588#  1579#  1580#  2259#  2263#  3825#  3826#  3871#  3875#  3879#  3883#
$SAVRE= ****** U         4789
$SCOPE  016342           1011   4553#
$SETUP= 000127           1002#  1010   1011   1013   1015   1017   1018   1019   1021   1048   1051   4182   4294
                         4428   4500   4526   4534   4554
$SSKO = 000253           1579#  1580   2259#  2263   3825#  3826   3871#  3875   3879#  3883
$STUP = 177777           1002#
$SVLAD  016550           4564   4593#
$SVPC = 000430            856#   861
$SWR  = 167400            588#   599    604    605    606    607    608    609    610    922    923    924    1018
                         1019   1021   1022   1123   1202   1283   1360   1432   1460   1593   1657   1719   1783
                         1966   2093   2239   2372   2595   2809   2859   3215   4177   4183   4198   4204   4206
                         4491   4492   4493   4494   4495   4504   4511   4523   4527   4539   4545   4546   4547
                         4548   4549   4555   4567   4569   4570   4573   4574   4575   4582   4583   4584   4596
                         4599   4602
$SWREG  001216            945#  1042
$SWRMK= 000000            610    611    4549   4550   4571
$TAGLE= 177777            588#  1107#  1110#  1111#  1114#  1138#  1149#  1153#  1161#  1208#  1211#  1220#  1224#
                         1236#  1240#  1253#  1257#  1272#  1276#  1295#  1299#  1311#  1315#  1328#  1332#  1347#
                         1351#  1369#  1373#  1385#  1389#  1402#  1406#  1421#  1425#  1441#  1450#  1466#  1469#
                         1475#  1503#  1508#  1525#  1532#  1533#  1546#  1550#  1552#  1556#  1561#  1567#  1570#
                         1571#  1574#  1576   1579#  1602#  1605#  1629#  1633#  1646#  1650#  1666#  1669#  1692#
                         1697#  1708#  1713#  1728#  1731#  1754#  1759#  1772#  1777#  1789#  1792#  1795#  1796#
                         1799#  1802#  1803#  1806#  1831#  1835#  1855#  1863#  1872#  1882#  1892#  1900#  1924#
                         1928#  1933#  1942#  1947#  1952#  2020#  2024#  2037#  2041#  2101#  2103#  2156#  2160#
                         2172#  2176#  2210#  2214#  2247#  2251#  2258#  2260#  2263#  2277#  2283#  2299#  2305#
                         2314#  2318#  2325#  2328#  2331#  2337#  2340#  2341#  2347#  2350#  2351#  2355#  2362#
                         2363#  2365#  2366#  2380#  2382#  2424#  2428#  2429#  2432#  2448#  2449#  2464#  2467#
                         2470#  2475#  2476#  2478#  2483#  2484#  2486#  2491#  2492#  2496#  2497#  2498#  2501#
                         2502#  2506#  2508#  2526#  2530#  2554#  2563#  2564#  2569#  2573#  2574#  2578#  2580#
                         2603#  2606#  2607#  2610#  2613#  2615#  2638#  2642#  2648#  2652#  2656#  2659#  2663#
                         2667#  2671#  2675#  2678#  2679#  2682#  2685#  2687#  2689#  2701#  2704#  2708#  2711#
                         2715#  2720#  2724#  2730#  2750#  2754#  2783#  2787#  2793#  2797#  2813#  2816#  2819#
                         2820#  2823#  2827#  2829#  2838#  2843#  2848#  2868#  2871#  2884#  2891#  2906#  2909#
                         2935#  2939#  2941#  2953#  2958#  2961#  2963#  2965#  3024#  3032#  3227#  3230#  3240#
                         3253#  3272#  3275#  3290#  3291#  3294#  3304#  3309#  3312#  3314#  3316#  3342#  3345#
                         3351#  3352#  3357#  3361#  3372#  3373#  3381#  3384#  3408#  3411#  3433#  3436#  3439#
                         3444#  3447#  3456#  3462#  3470#  3475#  3479#  3481#  3483#  3492#  3496#  3500#  3504#
                         3524#  3532#  3533#  3537#  3572#  3580#  3581#  3585#  3620#  3628#  3629#  3633#  3668#
                         3676#  3677#  3681#  3785#  3789#  3793#  3794#  3797#  3803#  3807#  3810   3825#  3835#
                         3838#  3870#  3872#  3875#  3878#  3880#  3883#  3884#  3885#  3887#  3888#
$TAGNU= 000257            588#  1106   1107#  1109   1111#  1137   1138#  1148   1149#  1207   1208#  1219   1220#
                         1235   1236#  1252   1253#  1271   1272#  1294   1295#  1310   1311#  1327   1328#  1346
                         1347#  1368   1369#  1384   1385#  1401   1402#  1420   1421#  1440   1441#  1465   1466#
                         1474   1475#  1502   1503#  1524   1525#  1531   1533#  1545   1546#  1555   1556#  1560
                         1561#  1569   1571#  1598   1600   1601   1602#  1628   1629#  1645   1646#  1662   1664
```

1665 1666# 1691 1692# 1707 1708# 1724 1726 1727 1728# 1753 1754# 1771
1772# 1788 1789# 1791 1792# 1794 1796# 1801 1803# 1830 1831# 1854 1855#
1871 1872# 1891 1892# 1923 1924# 1932 1933# 1946 1947# 2019 2020# 2036
2037# 2097 2099 2100 2101# 2155 2156# 2171 2172# 2209 2210# 2243 2245
2246 2247# 2257 2258# 2259# 2262 2263# 2276 2277# 2298 2299# 2313 2314#
2324 2325# 2327 2328# 2330 2331# 2339 2341# 2349 2351# 2352 2354 2355#
2376 2378 2379 2380# 2423 2424# 2427 2429# 2447 2448# 2450 2453 2454#
2463 2464# 2466 2467# 2469 2470# 2474 2476# 2477 2478# 2482 2484# 2485
2486# 2490 2492# 2500 2502# 2523 2525 2526# 2550 2552 2553 2554# 2562
2564# 2568 2569# 2572 2574# 2599 2601 2602 2603# 2605 2607# 2609 2610#
2637 2638# 2647 2648# 2655 2656# 2662 2663# 2666 2667# 2670 2671# 2677
2679# 2681 2682# 2700 2701# 2707 2708# 2714 2715# 2723 2724# 2749 2750#
2782 2783# 2792 2793# 2812 2813# 2815 2816# 2818 2820# 2822 2823# 2842
2843# 2867 2868# 2883 2884# 2905 2906# 2908 2909# 2934 2935# 2952 2953#
2957 2958# 3023 3024# 3226 3227# 3239 3240# 3271 3272# 3274 3275# 3289
3291# 3303 3304# 3308 3309# 3341 3342# 3344 3345# 3350 3352# 3371 3372#
3380 3381# 3407 3408# 3410 3411# 3432 3433# 3435 3436# 3443 3444# 3457
3460 3461# 3469 3470# 3474 3475# 3484 3487 3488# 3491 3492# 3499 3500#
3523 3524# 3531 3533# 3571 3572# 3579 3581# 3619 3620# 3627 3629# 3667
3668# 3675 3677# 3773# 3784 3785# 3788 3789# 3792 3794# 3802 3803# 3834
3835# 3864# 3869 3870# 3871# 3874 3875# 3877 3878# 3879# 3882 3883# 4154#
$TEMP = 000300 1079# 1080# 1082# 1083# 1085# 1086# 1087# 1089# 1090# 1092# 1093# 1095# 1096#
1097# 1103# 1104# 1109# 1110# 1112# 1113# 1114# 1126# 1127# 1129# 1130# 1131#
1132# 1133# 1134# 1135# 1136# 1139# 1140# 1142# 1143# 1153# 1156# 1157# 1158#
1160# 1161# 1162 1181# 1182# 1183# 1184# 1186# 1187# 1188# 1189# 1190# 1211#
1215# 1216# 1224# 1229# 1230# 1231# 1232# 1240# 1245# 1246# 1248# 1249# 1257#
1262# 1263# 1265# 1266# 1276# 1290# 1291# 1299# 1304# 1305# 1306# 1307# 1315#
1320# 1321# 1323# 1324# 1332# 1337# 1338# 1340# 1341# 1351# 1364# 1365# 1373#
1378# 1379# 1380# 1381# 1389# 1394# 1395# 1397# 1398# 1406# 1411# 1412# 1414#
1415# 1425# 1436# 1437# 1450# 1469# 1472# 1473# 1477# 1478# 1479# 1480# 1487#
1488# 1508# 1518# 1519# 1527# 1528# 1531# 1532# 1550# 1552# 1557# 1558# 1565#
1566# 1567# 1569# 1570# 1572# 1573# 1574# 1576# 1577 1579# 1580# 1605# 1608#
1609# 1612# 1613# 1633# 1638# 1639# 1650# 1669# 1672# 1673# 1677# 1678# 1697#
1704# 1705# 1713# 1731# 1734# 1735# 1739# 1740# 1759# 1767# 1768# 1777# 1794#
1795# 1799# 1801# 1802# 1806# 1811# 1812# 1817# 1818# 1835# 1839# 1840# 1849#
1850# 1863# 1868# 1869# 1882# 1886# 1887# 1900# 1904# 1905# 1911# 1912# 1928#
1942# 1952# 1970# 1971# 1974# 1975# 1977# 1978# 1980# 1981# 1982# 1983# 1984#
1985# 1986# 1987# 1999# 2000# 2006# 2007# 2024# 2041# 2045# 2046# 2048# 2049#
2051# 2052# 2055# 2056# 2075# 2076# 2077# 2078# 2080# 2081# 2082# 2083# 2084#
2103# 2108# 2109# 2110# 2111# 2112# 2113# 2115# 2116# 2117# 2118# 2120# 2121#
2127# 2128# 2141# 2142# 2160# 2176# 2186# 2187# 2190# 2191# 2193# 2194# 2196#
2197# 2214# 2223# 2224# 2225# 2226# 2227# 2228# 2230# 2231# 2232# 2233# 2234#
2251# 2260# 2263# 2283# 2287# 2288# 2305# 2309# 2310# 2318# 2337# 2339# 2340#
2347# 2349# 2350# 2355# 2362# 2363# 2365# 2366# 2382# 2403# 2404# 2406# 2407#
2408# 2409# 2412# 2413# 2414# 2415# 2418# 2419# 2420# 2421# 2425# 2426# 2427#
2428# 2430# 2431# 2432# 2439# 2440# 2442# 2443# 2449# 2452 2458# 2459# 2460#
2461# 2474# 2475# 2478# 2482# 2483# 2486# 2490# 2491# 2496# 2497# 2498# 2500#
2501# 2506# 2508# 2528# 2529# 2530# 2533# 2534# 2546# 2547# 2556# 2557# 2558#
2559# 2560# 2561# 2562# 2563# 2565# 2566# 2570# 2571# 2572# 2573# 2576# 2577#
2578# 2580# 2605# 2606# 2613# 2615# 2617# 2618# 2620# 2621# 2623# 2624# 2626#
2627# 2642# 2644# 2645# 2650# 2651# 2652# 2657# 2658# 2659# 2673# 2674# 2675#
2677# 2678# 2683# 2684# 2685# 2687# 2689# 2704# 2711# 2720# 2730# 2733# 2734#
2736# 2737# 2754# 2758# 2759# 2768# 2769# 2771# 2772# 2787# 2797# 2818# 2819#
2825# 2826# 2827# 2829# 2838# 2844# 2845# 2846# 2847# 2848# 2865# 2866# 2869#
2870# 2871# 2872 2879# 2880# 2881# 2882# 2885# 2886# 2887# 2888# 2889# 2890#
2891# 2892 2899# 2900# 2901# 2902# 2903# 2904# 2911# 2912# 2913# 2914# 2916#

		2917#	2919#	2920#	2922#	2923#	2939#	2941#	2944#	2945#	2947#	2948#	2949#	2950#
		2959#	2960#	2961#	2963#	2965#	2966	2979#	2980#	2989#	2990#	2992#	2993#	2994#
		2995#	2997#	2998#	3032#	3045#	3046#	3048#	3049#	3050#	3051#	3053#	3054#	3062#
		3063#	3065#	3066#	3067#	3068#	3070#	3071#	3078#	3079#	3081#	3082#	3083#	3084#
		3086#	3087#	3094#	3095#	3097#	3098#	3099#	3100#	3102#	3103#	3110#	3111#	3113#
		3114#	3115#	3116#	3118#	3119#	3126#	3127#	3129#	3130#	3131#	3132#	3134#	3135#
		3142#	3143#	3145#	3146#	3147#	3148#	3150#	3151#	3158#	3159#	3161	3162#	3163#
		3164#	3166#	3167#	3219#	3220#	3224#	3225#	3228#	3229#	3230#	3231	3235#	3236#
		3237#	3238#	3241#	3242#	3243#	3244#	3245#	3246#	3247#	3248#	3249#	3250#	3251#
		3252#	3253#	3254	3261#	3262#	3264#	3265#	3266#	3267#	3269#	3270#	3277#	3278#
		3280#	3281#	3282#	3283#	3285#	3286#	3287#	3288#	3289#	3290#	3292#	3293#	3294#
		3296#	3297#	3300#	3301#	3310#	3311#	3312#	3314#	3316#	3317	3321#	3322#	3323#
		3324#	3325#	3326#	3327#	3328#	3333#	3334#	3336#	3337#	3339#	3340#	3346#	3347#
		3348#	3349#	3350#	3351#	3353#	3354#	3355#	3356#	3357#	3359#	3360#	3361#	3362
		3366#	3367#	3373#	3374	3378#	3379#	3382#	3383#	3384#	3385	3394#	3395#	3396#
		3397#	3398#	3399#	3400#	3401#	3403#	3404#	3405#	3406#	3412#	3413#	3415#	3416#
		3418#	3419#	3421#	3422#	3424#	3425#	3426#	3427#	3429#	3430#	3437#	3438#	3439#
		3445#	3446#	3447#	3450#	3451#	3453#	3454#	3456#	3459	3462#	3464#	3465#	3466#
		3467#	3477#	3478#	3479#	3481#	3483#	3486	3496#	3504#	3516#	3517#	3518#	3519#
		3520#	3521#	3526#	3527#	3529#	3530#	3531#	3532#	3535#	3536#	3537#	3544#	3545#
		3546#	3547#	3548#	3549#	3550#	3551#	3553#	3554#	3557#	3558#	3564#	3565#	3566#
		3567#	3568#	3569#	3574#	3575#	3577#	3578#	3579#	3580#	3583#	3584#	3585#	3592#
		3593#	3594#	3595#	3596#	3597#	3598#	3599#	3601#	3602#	3605#	3606#	3612#	3613#
		3614#	3615#	3616#	3617#	3622#	3623#	3625#	3626#	3627#	3628#	3631#	3632#	3633#
		3640#	3641#	3642#	3643#	3644#	3645#	3646#	3647#	3649#	3650#	3653#	3654#	3660#
		3661#	3662#	3663#	3664#	3665#	3670#	3671#	3673#	3674#	3675#	3676#	3679#	3680#
		3681#	3688#	3689#	3690#	3691#	3692#	3693#	3694#	3695#	3697#	3698#	3701#	3702#
		3774#	3775#	3776#	3777#	3778#	3779#	3790#	3791#	3792#	3793#	3795#	3796#	3797#
		3805#	3806#	3807#	3810#	3811	3813	3823#	3824#	3825#	3826#	3838#	3849#	3866#
		3867#	3872#	3875#	3880#	3883#	3884#	3885#	3887#	3888#	3891#	3902#	3903#	3904#
		3905#	3906#	3907#	3922#	3923#	3924#	3925#	3927#	3928#	3929#	3930#	3931#	4052#
		4053#	4167#											
$TESTN	001200	936#	1124*	1203*	1284*	1361*	1433*	1461*	1594*	1658*	1720*	1784*	1967*	2094*
		2240*	2373*	2596*	2860*	3216*	4156	4594*						
$TIMES	001160	922#	1018*	1123*	1202*	1283*	1360*	1432*	1460*	1593*	1657*	1719*	1783*	1966*
		2093*	2239*	2372*	2595*	2809*	2859*	3215*	4183*	4582*	4589	4592*	4602	
$TKB	001146	915#	4292	4303	4320	4374	4380							
$TKS	001144	914#	4030	4292	4301	4317	4341*	4372	4378					
$TN =	000023	588#	599	1115	1123#	1124	1196	1202#	1203	1209	1279	1283#	1284	1354
		1360#	1361	1428	1432#	1433	1454	1460#	1461	1467	1582	1587	1593#	1594
		1603	1653	1657#	1658	1667	1715	1719#	1720	1729	1779	1783#	1784	1797
		1804	1861	1880	1898	1940	1955	1957	1966#	1967	2086	2093#	2094	2235
		2239#	2240	2249	2281	2303	2335	2345	2360	2368	2372#	2373	2510	2587
		2595#	2596	2802	2805	2809#	2852	2859#	2860	3035	3202	3215#	3216	
$TPB	001152	917#	4278*	4289										
$TPFLG	001157	921#	4227	4289										
$TPS	001150	916#	4276	4289										
$TRAP	017274	1015	4755#											
$TRAP2	017316	4766#	4777											
$TRP =	000012	4770#	4779#	4780#	4781#	4782#	4783#	4784	4785#	4786	4787#	4788#	4789#	
$TRPAD	017330	476C	4777#											
$TSK0 =	000251	1107#	1110	1111#	1114	1138#	1161	1208#	1211	1220#	1224	1236#	1240	1253#
		1257	1272#	1276	1295#	1299	1311#	1315	1328#	1332	1347#	1351	1369#	1373
		1385#	1389	1402#	1406	1421#	1425	1441#	1450	1466#	1469	1475#	1579	1602#
		1605	1629#	1633	1646#	1650	1666#	1669	1692#	1697	1708#	1713	1728#	1731
		1754#	1759	1772#	1777	1789#	1802	1803#	1806	1831#	1835	1855#	1863	1872#

```
                    1882    1892#   1900    1924#   1928    1933#   1942    1947#   1952    2020#   2024    2037#   2041
                    2101#   2103    2156#   2160    2172#   2176    2210#   2214    2247#   2251    2258#   2260    2263#
                    2366    2380#   2382    2424#   2428    2429#   2432    2448#   2449    2464#   2508    2526#   2530
                    2554#   2563    2564#   2580    2603#   2606    2607#   2615    2638#   2642    2648#   2652    2656#
                    2659    2663#   2689    2701#   2704    2708#   2711    2715#   2720    2724#   2730    2750#   2754
                    2783#   2787    2793#   2797    2813#   2838    2843#   2848    2868#   2871    2884#   2891    2906#
                    2965    3024#   3032    3227#   3230    3240#   3253    3272#   3316    3342#   3361    3372#   3373
                    3381#   3384    3408#   3483    3492#   3496    3500#   3504    3524#   3532    3533#   3537    3572#
                    3580    3581#   3585    3620#   3628    3629#   3633    3668#   3676    3677#   3681    3785#   3825
                    3835#   3838    3870#   3872    3875#   3888
$TSK1 = 000250      1149#   1153    1475#   1576    1579    1792#   1795    1796#   1799    2263#   2365    2467#   2501
                    2502#   2506    2569#   2573    2574#   2578    2610#   2613    2667#   2678    2679#   2687    2816#
                    2819    2820#   2829    2909#   2941    2953#   2963    3275#   3290    3291#   3294    3304#   3314
                    3345#   3351    3352#   3357    3411#   3462    3470#   3481    3785#   3810    3825    3875#   3887
$TSK2 = 000254      1503#   1508    1525#   1532    1533#   1552    1556#   1570    1571#   1574    2277#   2283    2299#
                    2305    2314#   2318    2325#   2350    2351#   2363    2470#   2475    2476#   2498    2671#   2675
                    2682#   2685    2823#   2827    2935#   2939    2958#   2961    3309#   3312    3433#   3456    3475#
                    3479    3789#   3793    3794#   3797    3803#   3807    3878#   3880    3883#   3885
$TSK3 - 000253      1546#   1550    1561#   1567    2328#   2340    2341#   2347    2355#   2362    2478#   2483    2484#
                    2497    3436#   3439    3444#   3447    3883#   3884
$TSK4 = 000126      2331#   2337    2486#   2491    2492#   2496
$TSTM   001004       881#
$TSTNM  001102       894#   1112*   4182*   4503    4539    4544    4571    4593*   4594    4599    4603
$TTYIN  015630      4399    4400    4417    4421#
$TYPBN= ****** U    4783
$TYPDS  016622      4615#   4782
$TYPE   014636      4227#   4459    4770    4778
$TYPEC  015050      4257    4264    4271    4276#   4277    4343
$TYPEX  015116      4282    4284    4287#
$TYPOC  017072      4700#   4779
$TYPON  017106      4699    4702#   4781
$TYPOS  017046      4695#   4780
$UNIT   001206       939#   1070*   2846*   4025    4070    4074*   4078    4087*   4098*
$UNITM  001010       883#
$USWR   001220       946#   1206    1597    1661    1723    1790    2096    2242    2326    2351    2375    2422    2598
                    2600    2661    2665    3343    3937    3941    4023    4096
$VECT1  001244       971#   4068
$VECT2  001246       972#
$XTSTR  016354      4558#
$YESNO= 000001      1088#   1089#   1091#   1092#   1094#   1095#   1103#   1104#   1156#   1157#   1159#   1160#   1185#
                    1186#   1231#   1232#   1248#   1249#   1265#   1266#   1306#   1307#   1323#   1324#   1340#   1341#
                    1380#   1381#   1397#   1398#   1414#   1415#   1557#   1558#   1704#   1705#   1977#   1978#   1999#
                    2000#   2006#   2007#   2048#   2049#   2079#   2080#   2120#   2121#   2141#   2142#   2190#   2191#
                    2229#   2230#   2259#   2260#   2439#   2440#   2442#   2443#   2458#   2459#   2460#   2461#   2533#
                    2534#   2558#   2559#   2560#   2561#   2565#   2566#   2570#   2571#   2576#   2577#   2623#   2624#
                    2650#   2651#   2657#   2658#   2673#   2674#   2683#   2684#   2736#   2737#   2758#   2759#   2825#
                    2826#   2846#   2847#   2889#   2890#   2911#   2912#   2913#   2914#   2916#   2917#   2922#   2923#
                    2944#   2945#   2947#   2948#   2949#   2950#   2959#   2960#   3050#   3051#   3053#   3054#   3067#
                    3068#   3070#   3071#   3083#   3084#   3086#   3087#   3099#   3100#   3102#   3103#   3115#   3116#
                    3118#   3119#   3131#   3132#   3134#   3135#   3147#   3148#   3150#   3151#   3163#   3164#   3166#
                    3167#   3245#   3246#   3251#   3252#   3277#   3278#   3280#   3281#   3282#   3283#   3285#   3286#
                    3287#   3288#   3292#   3293#   3296#   3297#   3300#   3301#   3310#   3311#   3359#   3360#   3382#
                    3383#   3415#   3416#   3421#   3422#   3426#   3427#   3437#   3438#   3445#   3446#   3450#   3451#
                    3453#   3454#   3464#   3465#   3466#   3467#   3477#   3478#   3518#   3519#   3526#   3527#   3529#
                    3530#   3535#   3536#   3546#   3547#   3550#   3551#   3553#   3554#   3557#   3558#   3566#   3567#
                    3574#   3575#   3577#   3578#   3583#   3584#   3594#   3595#   3598#   3599#   3601#   3602#   3605#
```

```
                    3606#  3614#  3615#  3622#  3623#  3625#  3626#  3631#  3632#  3642#  3643#  3646#  3647#
                    3649#  3650#  3653#  3654#  3662#  3663#  3670#  3671#  3673#  3674#  3679#  3680#  3690#
                    3691#  3694#  3695#  3697#  3705#  3701#  3702#  3823#  3824#  3871#  3872#  3879#  3880#
                    3926#  3927#  4052#  4053#
$$ARGC= 000000      3773#  3864#  4154#
$$BYTE= 000402      1105#  1147#  1206#  1218#  1234#  1251#  1270#  1293#  1309#  1326#  1345#  1367#  1383#
                    1400#  1419#  1439#  1464#  1523#  1554#  1559#  1597#  1599#  1644#  1661#  1663#  1706#
                    1723#  1725#  1770#  1787#  1790#  1853#  1870#  1890#  1931#  1945#  2035#  2096#  2098#
                    2170#  2242#  2244#  2312#  2323#  2326#  2329#  2351#  2353#  2375#  2377#  2422#  2462#
                    2465#  2468#  2476#  2484#  2522#  2524#  2549#  2551#  2567#  2598#  2600#  2608#  2646#
                    2654#  2661#  2665#  2669#  2680#  2699#  2706#  2713#  2722#  2791#  2811#  2814#  2821#
                    2841#  2907#  2951#  2956#  3022#  3273#  3302#  3307#  3343#  3409#  3434#  3442#  3468#
                    3473#  3490#  3498#  3522#  3570#  3618#  3666#  3787#  3801#  3833#
$$FLAG= 000001      1105#  1107   1114#  1147#  1149   1153#  1206#  1208   1211#  1218#  1220   1224#  1234#
                    1236   1240#  1251#  1253   1257#  1270#  1272   1276#  1293#  1295   1299#  1309#  1311
                    1315#  1326#  1328   1332#  1345#  1347   1351#  1367#  1369   1373#  1383#  1385   1389#
                    1400#  1402   1406#  1419#  1421   1425#  1439#  1441   1450#  1464#  1466   1469#  1502#
                    1508#  1523#  1525   1545#  1550#  1552#  1554#  1556   1559#  1561   1567#  1574#  1597#
                    1599#  1602   1605#  1628#  1633#  1644#  1646   1650#  1661#  1663#  1666   1669#  1691#
                    1697#  1706#  1708   1713#  1723#  1725#  1728   1731#  1753#  1759#  1770#  1772   1777#
                    1787#  1789   1790#  1792   1799#  1806#  1830#  1835#  1853#  1855   1863#  1870#  1872
                    1882#  1890#  1892   1900#  1923#  1928#  1931#  1933   1942#  1945#  1947   1952#  2019#
                    2024#  2035#  2037   2041#  2096#  2098#  2101   2103#  2155#  2160#  2170#  2172   2176#
                    2209#  2214#  2242#  2244#  2247   2251#  2276#  2283#  2298#  2305#  2312#  2314   2318#
                    2323#  2325   2326#  2328   2329#  2331   2337#  2347#  2351#  2353#  2355   2362#  2375#
                    2377#  2380   2382#  2422#  2424   2432#  2462#  2464   2465#  2467   2468#  2470   2476#
                    2478   2484#  2486   2496#  2506#  2508#  2522#  2524#  2526   2530#  2549#  2551#  2554
                    2567#  2569   2578#  2580#  2598#  2600#  2603   2608#  2610   2613#  2615#  2637#  2642#
                    2646?  2648   2652#  2654#  2656   2659#  2661#  2663   2665#  2667   2669#  2671   2675#
                    2680#  2682   2685#  2687#  2689#  2699#  2701   2704#  2706#  2708   2711#  2713#  2715
                    2720#  2722#  2724   2730#  2749#  2754#  2782#  2787#  2791#  2793   2797#  2811#  2813
                    2814#  2816   2821#  2823   2827#  2829#  2838#  2841#  2843   2848#  2907#  2909   2934#
                    2939#  2941#  2951#  2953   2956#  2958   2961#  2963#  3022#  3024   3032#  3273#  3275
                    3294#  3302#  3304   3307#  3309   3312#  3314#  3343#  3345   3357#  3409#  3411   3434#
                    3436   3439#  3442#  3444   3447#  3462#  3468#  3470   3473#  3475   3479#  3481#  3490#
                    3492   3496#  3498#  3500   3504#  3522#  3524   3537#  3570#  3572   3585#  3618#  3620
                    3633#  3666#  3668   3681#  3787#  3789   3797#  3801#  3803   3807#  3833#  3835   3838#
$$FROM= 000000      1492#  1537#  1618#  1682#  1744#  1820#  1842#  1913#  1989#  2009#  2028#  2058#  2132#
                    2145#  2163#  2200#  2268#  2290#  2629#  2739#  2760#  2774#  2925#  2954#  3305#  3471#
                    3818#
$$GET4= 000000      4198#
$$LOC = 012710      1106#  1107   1148#  1149   1162#  1163   1207#  1208   1219#  1220   1235#  1236   1252#
                    1253   1271#  1272   1294#  1295   1310#  1311   1327#  1328   1346#  1347   1368#  1369
                    1384#  1385   1401#  1402   1420#  1421   1440#  1441   1465#  1466   1502#  1503   1524#
                    1525   1545#  1546   1555#  1556   1560#  1561   1600#  1601   1628#  1629   1645#  1646
                    1664#  1665   1691#  1692   1707#  1708   1726#  1727   1753#  1754   1771#  1772   1788#
                    1789   1791#  1792   1830#  1831   1854#  1855   1871#  1872   1891#  1892   1923#  1924
                    1932#  1933   1946#  1947   2019#  2020   2036#  2037   2099#  2100   2155#  2156   2171#
                    2172   2209#  2210   2245#  2246   2276#  2277   2298#  2299   2313#  2314   2324#  2325
                    2327#  2328   2330#  2331   2352#  2353   2354#  2355   2378#  2379   2423#  2424   2452#
                    2453   2463#  2464   2466#  2467   2469#  2470   2477#  2478   2485#  2486   2523#  2524
                    2525#  2526   2552#  2553   2568#  2569   2601#  2602   2609#  2610   2637#  2638   2647#
                    2648   2655#  2656   2662#  2663   2666#  2667   2670#  2671   2681#  2682   2700#  2701
                    2707#  2708   2714#  2715   2723#  2724   2749#  2750   2782#  2783   2792#  2793   2812#
                    2813   2815#  2816   2822#  2823   2842#  2843   2872#  2873   2892#  2893   2908#  2909
                    2934#  2935   2952#  2953   2957#  2958   2966#  2967   3023#  3024   3231#  3232   3254#
```

```
                    3255    3274#   3275    3303#   3304    3308#   3309    3317#   3318    3344#   3345    3362#   3363
                    3374#   3375    3385#   3386    3410#   3411    3435#   3436    3443#   3444    3459#   3460    3469#
                    3470    3474#   3475    3486#   3487    3491#   3492    3499#   3500    3523#   3524    3571#   3572
                    3619#   3620    3667#   3668    3788#   3789    3802#   3803    3834#   3835
$$LOCN= 000000      3773#   3852    3864#   3893    4154#   4169
$$RETU= 000000      3773#   3836#   3840#   3850    3864#   3892    4154#   4168
$$RTN1= 000255      3773#   3836    3849    3864#   3891    4154#   4167
$$RTN2= 000256      3773#   3841    3851    3864#   3892    4154#   4168
$$TO  = 000001      1492#   1498    1537#   1543    1618#   1624    1682#   1688    1744#   1750    1820#   1826    1842#
                    1845    1913#   1919    1989#   1995    2009#   2015    2028#   2031    2058#   2064    2132#   2138
                    2145#   2151    2163#   2166    2200#   2206    2268#   2274    2290#   2296    2629#   2635    2739#
                    2745    2760#   2763    2774#   2780    2925#   2931    2954#   2955    3305#   3306    3471#   3472
                    3818#   3821
$OFILL  017271      4696*   4700*   4710    4745#
$1      002062      1106    1110#
$10     002430      1252    1257#
$100    005272      2276    2283#
$101    005334      2298    2305#
$102    005350      2313    2318#
$103    005406      2324    2350#
$104    005400      2327    2340#
$105    005376      2330    2337#
$106    005404      2339    2347#
$107    005430      2349    2363#
$11     002460      1271    1276#
$110    005430      2352    2354    2362#
$111    005470      2376    2379#
$112    005474      2378    2382#
$113    005562      2423    2428#
$114    005570      2427    2432#
$115    005606      2447#   2452
$116    005624      2450    2453#
$117    005730      2463    2508#
$12     002530      1294    1299#
$120    005726      2466    2501#
$121    005672      2469    2475#
$122    005724      2474    2498#
$123    005706      2477    2483#
$124    005724      2482    2497#
$125    005722      2485    2491#
$126    005724      2490    2496#
$127    005730      2500    2506#
$13     002556      1310    1315#
$130    005762      2523    2525    2530#
$131    006022      2550    2553#
$132    006044      2552    2563#
$133    006076      2562    2580#
$134    006070      2568    2573#
$135    006076      2572    2578#
$136    006140      2599    2602#
$137    006146      2601    2606#
$14     002604      1327    1332#
$140    006162      2605    2615#
$141    006162      2609    2613#
$142    006244      2637    2642#
$143    006262      2647    2652#
```

$144	006276	2655	2659#
$145	006350	2662	2689#
$146	006334	2666	2678#
$147	006332	2670	2675#
$15	002634	1346	1351#
$150	006350	2677	2687#
$151	006350	2681	2685#
$152	006376	2700	2704#
$153	006410	2707	2711#
$154	006422	2714	2720#
$155	006434	2723	2730#
$156	006504	2749	2754#
$157	006574	2782	2787#
$16	002672	1368	1373#
$160	006606	2792	2797#
$161	007062	2812	2838#
$162	006706	2815	2819#
$163	006760	2818	2829#
$164	006760	2822	2827#
$165	007106	2842	2848#
$166	007134	2867#	2872
$167	007166	2883#	2892
$17	002720	1384	1389#
$170	007224	2905#	2966
$171	007314	2908	2941#
$172	007314	2934	2939#
$173	007356	2952	2963#
$174	007356	2957	2961#
$175	007534	3023	3032#
$176	010136	3226#	3231
$177	010156	3239#	3254
$2	002066	1109	1114#
$20	002746	1401	1406#
$200	010244	3271#	3317
$201	010302	3274	3290#
$202	010306	3289	3294#
$203	010344	3303	3314#
$204	010344	3308	3312#
$205	010406	3341#	3362
$206	010430	3344	3351#
$207	010440	3350	3357#
$21	002776	1420	1425#
$210	010472	3371#	3374
$211	010506	3380#	3385
$212	010562	3407#	3486
$213	010666	3410	3462#
$214	010622	3432#	3459
$215	010636	3435	3439#
$216	010646	3443	3447#
$217	010666	3457	3460#
$22	003036	1440	1450#
$220	010726	3469	3481#
$221	010726	3474	3479#
$222	010744	3484	3487#
$223	010756	3491	3496#
$224	010766	3499	3504#

```
$225    011040          3523    3532#
$226    011044          3531    3537#
$227    011160          3571    3580#
$23     003066          1465    1469#
$230    011164          3579    3585#
$231    011300          3619    3628#
$232    011304          3627    3633#
$233    011420          3667    3676#
$234    011424          3675    3681#
$235    012726          3836    3849#
$236    012730          3841    3851#
$237    012600          3784#   3825
$24     003072          1474#   1579
$240    012702          3811    3813    3826#
$241    012616          3788    3793#
$242    012624          3792    3797#
$243    012642          3802    3807#
$244    012714          3834    3838#
$245    013010          3891#
$246    013010          3892#
$247    012756          3869    3872#
$25     003320          1577    1580#
$250    012752          3870#   3887
$251    013002          3874    3888#
$252    012770          3877    3880#
$253    012766          3878#   3884
$254    013000          3882    3885#
$255    014512          4167#
$256    014512          4168#
$26     003146          1502    1508#
$27     003206          1524    1532#
$3      002134          1137#   1162
$30     003242          1531    1552#
$31     003242          1545    1550#
$32     003300          1555    1570#
$33     003276          1560    1567#
$34     003306          1569    1574#
$35     003366          1598    1601#
$36     003370          1600    1605#
$37     003436          1628    1633#
$4      002160          1148    1153#
$40     003460          1645    1650#
$40CAT= ****** U        4513    4555
$41     003516          1662    1665#
$42     003520          1664    1669#
$43     003566          1691    1697#
$44     003606          1707    1713#
$45     003644          1724    1727#
$46     003646          1726    1731#
$47     003714          1753    1759#
$5      002334          1207    1211#
$50     003732          1771    1777#
$51     003774          1788    1802#
$52     003770          1791    1795#
$53     003772          1794    1799#
$54     003776          1801    1806#
```

```
$55     004044          1830    1835#
$56     004102          1854    1863#
$57     004122          1871    1882#
$6      002354          1219    1224#
$60     004144          1891    1900#
$61     004212          1923    1928#
$62     004226          1932    1942#
$63     004240          1946    1952#
$64     004432          2019    2024#
$65     004460          2036    2041#
$66     004626          2097    2100#
$67     004632          2099    2103#
$7      002402          1235    1240#
$70     005002          2155    2160#
$71     005030          2171    2176#
$72     005110          2209    2214#
$73     005214          2243    2246#
$74     005216          2245    2251#
$75     005226          2257    2260#
$76     005224          2258#   2365
$77     005432          2262    2366#
.       = 017354         827#    831#    842#    856     857#    859#    861#    862#    868     869#    871#    873#    891#
                         928    1000#   1008    1021    1022    1064#   1099    1106    1107    1148    1149    1162    1163
                        1207    1208    1219    1220    1235    1236    1252    1253    1271    1272    1294    1295    1310
                        1311    1327    1328    1346    1347    1368    1369    1384    1385    1401    1402    1420    1421
                        1440    1441    1465    1466    1502    1503    1524    1525    1545    1546    1555    1556    1560
                        1561    1600    1601    1628    1629    1645    1646    1664    1665    1691    1692    1707    1708
                        1726    1727    1753    1754    1771    1772    1788    1789    1791    1792    1830    1831    1854
                        1855    1871    1872    1891    1892    1923    1924    1932    1933    1946    1947    2019    2020
                        2036    2037    2099    2100    2155    2156    2171    2172    2209    2210    2245    2246    2276
                        2277    2298    2299    2313    2314    2324    2325    2327    2328    2330    2331    2352    2353
                        2354    2355    2378    2379    2423    2424    2452    2453    2463    2464    2466    2467    2469
                        2470    2477    2478    2485    2486    2523    2524    2525    2526    2552    2553    2568    2569
                        2601    2602    2609    2610    2637    2638    2647    2648    2655    2656    2662    2663    2666
                        2667    2670    2671    2681    2682    2700    2701    2707    2708    2714    2715    2723    2724
                        2749    2750    2782    2783    2792    2793    2812    2813    2815    2816    2817#   2822    2823
                        2824#   2842    2843    2872    2873    2892    2893    2908    2909    2934    2935    2952    2953
                        2957    2958    2966    2967    3023    3024    3192#   3231    3232    3254    3255    3274    3275
                        3303    3304    3308    3309    3317    3318    3344    3345    3362    3363    3374    3375    3385
                        3386    3410    3411    3435    3436    3443    3444    3459    3460    3469    3470    3474    3475
                        3486    3487    3491    3492    3499    3500    3523    3524    3571    3572    3619    3620    3667
                        3668    3708#   3709#   3710#   3711#   3728#   3788    3789    3802    3803    3834    3835    3935
                        3975#   3985#   3987#   4155#   4160#   4162#   4206    4210    4289    4292    4421#   4422    4428
                        4480#   4539    4602    4603    4669#
.$ASTA= ****** U        4432    4435
.$X     = 001000         868#    873
```

```
BEGIN       1#
BGNSRV    588#  2519  2542  3041  3058  3074  3090  3106  3122  3138  3154  3513  3541  3561  3589
         3609  3637  3657  3685  4048
BGNSUB    588#  1138  1214  1228  1244  1261  1289  1303  1319  1336  1363  1377  1393  1410  1435
         1607  1637  1671  1733  1810  1867  1885  1903  1985  2044  2114  2185  2616  2732
BRESET    588#  1100  1267  1342  1416  1447  1640  2070  2218  2252  2434  2691  2798  2985  3388
CALL        1#  1491  1536  1617  1681  1743  1819  1842  1912  1988  2008  2028  2057  2131  2144
         2163  2199  2267  2289  2628  2738  2760  2773  2924  2953  3304  3470  3818
CASE        1#
CLRVEC    588#  1180  2074  2224  3921
COMMEN      1#   721#
DECR        1#
DECRU       1#
DEFAUL      1#
ELSE        1#  1108  1530  1568  1793  1800  2338  2348  2426  2473  2481  2489  2499  2561  2571
         2604  2676  2817  3288  3349  3530  3578  3626  3674  3791
END         1#
ENDCOM      1#   721#
ENDDEC      1#
ENDDO       1#
ENDIF       1#  1113  1152  1210  1223  1239  1256  1275  1298  1314  1331  1350  1372  1388  1405
         1424  1449  1468  1507  1549  1551  1566  1573  1604  1632  1649  1668  1696  1712  1730
         1758  1776  1798  1805  1834  1862  1881  1899  1927  1941  1951  2023  2040  2102  2159
         2175  2213  2250  2282  2304  2317  2336  2346  2361  2381  2431  2495  2505  2507  2529
         2577  2579  2612  2614  2641  2651  2658  2674  2684  2686  2688  2703  2710  2719  2729
         2753  2786  2796  2826  2828  2837  2847  2938  2940  2960  2962  3031  3293  3311  3313
         3356  3438  3446  3461  3478  3480  3495  3503  3536  3584  3632  3680  3796  3806  3837
ENDINC      1#  2364  3883  3886
ENDLOO      1#  1578  3824
ENDRTN      1#  3848  3890  4166
ENDSEL      1#
ENDSRV    588#  2534  2581  3054  3071  3087  3103  3119  3135  3151  3167  3538  3558  3586  3606
         3634  3654  3682  3702  4053
ENDSUB    588#  1154  1225  1241  1258  1277  1300  1316  1333  1352  1374  1390  1407  1426  1452
         1635  1651  1698  1760  1864  1883  1901  1953  2042  2072  2177  2215  2731  2800
ERRDF     588#  1150
ERRHRD    588#  1221  1237  1254  1273  1296  1312  1329  1348  1370  1386  1403  1422  1444  1504
         1547  1562  1630  1647  1693  1709  1755  1773  1832  1856  1874  1894  1925  1934  1948
         2021  2038  2068  2157  2173  2211  2278  2299  2315  2332  2342  2357  2471  2479  2487
         2493  2503  2639  2701  2708  2715  2724  2751  2784  2794  2936  3024  3493  3501
ERROR     615#  1151  1166  1170  1174  1178  1222  1238  1255  1274  1297  1313  1330  1349  1371
         1387  1404  1423  1445  1505  1548  1563  1631  1648  1694  1710  1756  1774  1833  1857
         1875  1895  1926  1935  1949  2022  2039  2069  2158  2174  2212  2279  2300  2316  2333
         2343  2358  2472  2480  2488  2494  2504  2640  2695  2702  2709  2716  2725  2752  2785
         2795  2937  3025  3494  3502
ESCAPE      1#   721#
EXIF        1#  1575
EXIFB       1#  3809
EXIT      588#  1208  1466  1581  1602  1666  1728  1796  1803  1860  1879  1897  1939  1954  2248
         2280  2302  2334  2344  2359  2509  2801  3034
GETPRI      1#   721#
GETSWR      1#   721#  1051#
IF          1#  1146  1205  1217  1233  1250  1269  1292  1308  1325  1344  1366  1382  1399  1418
         1438  1463  1522  1553  1558  1596  1643  1660  1705  1722  1769  1786  1789  1852  1869
         1889  1930  1944  2034  2095  2169  2241  2311  2322  2325  2328  2374  2421  2461  2464
         2467  2521  2548  2566  2597  2607  2653  2660  2664  2668  2679  2698  2705  2712  2721
```

```
                                                
```

SETVEC	588#	1128	1979	2107	3901										
SKIP	1#	721#	1209	1467	1582	1603	1667	1729	1797	1804	1861	1880	1898	1940	1955
	2249	2281	2303	2335	2345	2360	2510	2802	3035						
SLASH	1#	721#													
SPACE	721#														
STARS	1#	721#	734	736	754	756	774	776	795	797	816	824	854	865	867
	874	887	928	931	1115	1121	1191	1193	1196	1200	1279	1281	1354	1358	1428
	1430	1454	1458	1587	1591	1653	1655	1715	1717	1779	1781	1957	1964	2086	2091
	2235	2237	2368	2370	2516	2518	2539	2541	2587	2593	2805	2807	2852	2857	3037
	3039	3202	3213	3509	3511	3749	3768	3854	3861	4042	4047	4061	4065	4173	4212
	4291	4294	4362	4391	4430	4487	4541	4605	4672	4749					
STRUCT	1#	588#													
SWRSU	1#	721#	1023#												
TRMTRP	4770#														
TYPBIN	1#	721#													
TYPDEC	1#	721#	2835	4193											
TYPNAM	1#	721#	1044												
TYPNUM	1#	721#													
TYPOCS	1#	721#	4158												
TYPOCT	1#	721#	2831	2833	4155	4160	4162	4164	4312						
TYPTXT	1#	721#	2816	2823	2830	2832	2834	3934	3948	3974	3984	3986	4154	4156	4159
	4161	4163													
UNTIL	1#	1160	2448	2870	2890	2964	3229	3252	3315	3360	3372	3383	3455	3482	
UNTILB	1#														
WAITMS	588#	1841	2027	2162	2759	3817									
WHILE	1#														
WHILEB	1#														
$ADDON	1#	1105	1107	1111	1137	1138	1147	1149	1206	1208	1218	1220	1234	1236	1251
	1253	1270	1272	1293	1295	1309	1311	1326	1328	1345	1347	1367	1369	1383	1385
	1400	1402	1419	1421	1439	1441	1464	1466	1474	1475	1502	1503	1523	1525	1533
	1545	1546	1554	1556	1559	1561	1571	1579	1597	1602	1628	1629	1644	1646	1661
	1666	1691	1692	1706	1708	1723	1728	1753	1754	1770	1772	1787	1789	1790	1792
	1796	1803	1830	1831	1853	1855	1870	1872	1890	1892	1923	1924	1931	1933	1945
	1947	2019	2020	2035	2037	2096	2101	2155	2156	2170	2172	2209	2210	2242	2247
	2256	2258	2259	2263	2276	2277	2298	2299	2312	2314	2323	2325	2326	2328	2329
	2331	2341	2351	2355	2375	2380	2422	2424	2429	2447	2448	2454	2462	2464	2465
	2467	2468	2470	2476	2478	2484	2486	2492	2502	2522	2526	2549	2554	2564	2567
	2569	2574	2598	2603	2607	2608	2610	2637	2638	2646	2648	2654	2656	2661	2663
	2665	2667	2669	2671	2679	2680	2682	2699	2701	2706	2708	2713	2715	2722	2724
	2749	2750	2782	2783	2791	2793	2811	2813	2814	2816	2820	2821	2823	2841	2843
	2867	2868	2883	2884	2905	2906	2907	2909	2934	2935	2951	2953	2956	2958	3022
	3024	3226	3227	3239	3240	3271	3272	3273	3275	3291	3302	3304	3307	3309	3341
	3342	3343	3345	3352	3371	3372	3380	3381	3407	3408	3409	3411	3432	3433	3434
	3436	3442	3444	3461	3468	3470	3473	3475	3488	3490	3492	3498	3500	3522	3524
	3533	3570	3572	3581	3618	3620	3629	3666	3668	3677	3773	3784	3785	3787	3789
	3794	3801	3803	3825	3833	3835	3864	3868	3870	3871	3875	3876	3878	3879	3883
	4154														
$AND	1#	2351	2522												
$BRANC	1#	1106	1109	1148	1162	1207	1219	1235	1252	1271	1294	1310	1327	1346	1368
	1384	1401	1420	1440	1465	1502	1524	1531	1545	1555	1560	1569	1577	1579	1598
	1600	1628	1645	1662	1664	1691	1707	1724	1726	1753	1771	1788	1791	1794	1801
	1830	1854	1871	1891	1923	1932	1946	2019	2036	2097	2099	2155	2171	2209	2243
	2245	2257	2262	2276	2298	2313	2324	2327	2330	2339	2349	2352	2354	2365	2376
	2378	2423	2427	2450	2452	2463	2466	2469	2474	2477	2482	2485	2490	2500	2523
	2525	2550	2552	2562	2568	2572	2599	2601	2605	2609	2637	2647	2655	2662	2666
	2670	2677	2681	2700	2707	2714	2723	2749	2782	2792	2812	2815	2818	2822	2842

$BRCOD 1# 1576 1597 1661 1723 2096 2242 2261 2375 2449 2549 2598 3456 3483 3810
3812 3873 3881

$CALL 1# 1492 1537 1618 1682 1744 1820 1842 1913 1989 2009 2028 2058 2132 2145
2163 2200 2268 2290 2629 2739 2760 2774 2925 2954 3305 3471 3818

$CHECK 1# 1105 1147 1206 1218 1234 1251 1270 1293 1309 1326 1345 1367 1383 1400
1419 1439 1464 1523 1554 1559 1597 1644 1661 1706 1723 1770 1787 1790 1853
1870 1890 1931 1945 2035 2096 2170 2242 2312 2323 2326 2329 2351 2375 2422
2462 2465 2468 2476 2484 2522 2549 2567 2598 2608 2646 2654 2661 2665 2669
2680 2699 2706 2713 2722 2791 2811 2814 2821 2841 2907 2951 2956 3022 3273
3302 3307 3343 3409 3434 3442 3468 3473 3490 3498 3522 3570 3618 3666 3787
3801 3833

$CHK1 1# 1079 1082 1085 1096 1112 1126 1129 1130 1131 1132 1133 1135 1139 1142
1181 1182 1183 1186 1187 1188 1189 1215 1229 1245 1262 1290 1304 1320 1337
1364 1378 1394 1411 1436 1472 1477 1479 1487 1518 1527 1565 1572 1608 1612
1638 1672 1677 1734 1739 1767 1811 1817 1839 1849 1868 1886 1904 1911 1970
1974 1980 1981 1982 1983 1984 1986 2045 2051 2055 2075 2076 2077 2080 2081
2082 2083 2108 2109 2110 2111 2112 2115 2117 2127 2186 2193 2196 2223 2225
2226 2227 2230 2231 2232 2233 2256 2259 2287 2309 2403 2406 2408 2412 2414
2418 2420 2425 2430 2528 2546 2556 2617 2620 2626 2644 2733 2768 2771 2844
2865 2869 2879 2881 2885 2887 2899 2901 2903 2919 2979 2989 2992 2994 2997
3045 3048 3062 3065 3078 3081 3094 3097 3110 3113 3126 3129 3142 3145 3158
3161 3219 3224 3228 3235 3237 3241 3243 3247 3249 3261 3264 3266 3269 3321
3323 3325 3327 3333 3336 3339 3346 3348 3353 3355 3366 3378 3394 3396 3398
3400 3403 3405 3412 3418 3424 3429 3516 3520 3544 3548 3564 3568 3592 3596
3612 3616 3640 3644 3660 3664 3688 3692 3774 3776 3778 3790 3795 3805 3866
3868 3871 3876 3879 3902 3903 3904 3905 3906 3922 3923 3924 3927 3928 3929
3930

$CKOP2 1# 1087 1090 1093 1103 1156 1158 1184 1231 1248 1265 1306 1323 1340 1380
1397 1414 1557 1704 1977 1999 2006 2048 2078 2120 2141 2190 2228 2439 2442
2458 2460 2533 2558 2560 2565 2570 2576 2623 2650 2657 2673 2683 2736 2758
2825 2846 2889 2911 2913 2916 2922 2944 2947 2949 2959 3050 3053 3067 3070
3083 3086 3099 3102 3115 3118 3131 3134 3147 3150 3163 3166 3245 3251 3277
3280 3282 3285 3287 3292 3296 3300 3310 3359 3382 3415 3421 3426 3437 3445
3450 3453 3464 3466 3477 3518 3526 3529 3535 3546 3550 3553 3557 3566 3574
3577 3583 3594 3598 3601 3605 3614 3622 3625 3631 3642 3646 3649 3653 3662
3670 3673 3679 3690 3694 3697 3701 3823 3925 4052

$CKR6 1#

$CMND 1# 1105 1147 1206 1218 1234 1251 1270 1293 1309 1326 1345 1367 1383 1400
1419 1439 1464 1523 1554 1559 1597 1599 1644 1661 1663 1706 1723 1725 1770
1787 1790 1853 1870 1890 1931 1945 2035 2096 2098 2170 2242 2244 2312 2323
2326 2329 2351 2353 2375 2377 2422 2462 2465 2468 2476 2484 2522 2524 2549
2551 2567 2598 2600 2608 2646 2654 2661 2665 2669 2680 2699 2706 2713 2722
2791 2811 2814 2821 2841 2907 2951 2956 3022 3273 3302 3307 3343 3409 3434
3442 3468 3473 3490 3498 3522 3570 3618 3666 3787 3801 3833

$COMPA 1# 1105 1147 1206 1218 1234 1251 1270 1293 1309 1326 1345 1367 1383 1400
1419 1439 1464 1502 1523 1545 1554 1559 1597 1628 1644 1661 1691 1706 1723
1753 1770 1787 1790 1830 1853 1870 1890 1923 1931 1945 2019 2035 2096 2155
2170 2209 2242 2256 2276 2298 2312 2323 2326 2329 2351 2375 2422 2462 2465
2468 2476 2484 2522 2549 2567 2598 2608 2637 2646 2654 2661 2665 2669 2680
2699 2706 2713 2722 2749 2782 2791 2811 2814 2821 2841 2907 2934 2951 2956
3022 3273 3302 3307 3343 3409 3434 3442 3468 3473 3490 3498 3522 3570 3618
3666 3787 3801 3833 3868 3876

$COUNT	1#	1492	1537	1618	1682	1744	1820	1842	1913	1989	2009	2028	2058	2132	2145
	2163	2200	2268	2290	2629	2739	2760	2774	2925	2954	3305	3471	3818		
$DO	1#														
$ELSE	1#	2351	2476	2484											
$ERRMS	1#														
$EXIFA	1#														
$EXIFO	1#														
$EXIF2	1#	3810													
$EXIF3	1#														
$GENBR	1#	1106	1109	1148	1162	1207	1219	1235	1252	1271	1294	1310	1327	1346	1368
	1384	1401	1420	1440	1465	1502	1524	1531	1545	1555	1560	1569	1577	1579	1598
	1600	1628	1645	1662	1664	1691	1707	1724	1726	1753	1771	1788	1791	1794	1801
	1830	1854	1871	1891	1923	1932	1946	2019	2036	2097	2099	2155	2171	2209	2243
	2245	2257	2262	2276	2298	2313	2324	2327	2330	2339	2349	2352	2354	2365	2376
	2378	2423	2427	2450	2452	2463	2466	2469	2474	2477	2482	2485	2490	2500	2523
	2525	2550	2552	2562	2568	2572	2599	2601	2605	2609	2637	2647	2655	2662	2666
	2670	2677	2681	2700	2707	2714	2723	2749	2782	2792	2812	2815	2818	2822	2842
	2872	2892	2908	2934	2952	2957	2966	3023	3231	3254	3274	3289	3303	3308	3317
	3344	3350	3362	3374	3385	3410	3435	3443	3457	3459	3469	3474	3484	3486	3491
	3499	3523	3531	3571	3579	3619	3627	3667	3675	3788	3792	3802	3811	3813	3825
	3834	3836	3841	3869	3874	3877	3882	3884	3887						
$GENTA	1#	1110	1114	1137	1153	1211	1224	1240	1257	1276	1299	1315	1332	1351	1373
	1389	1406	1425	1450	1469	1474	1508	1532	1550	1552	1567	1570	1574	1580	1601
	1605	1633	1650	1665	1669	1697	1713	1727	1731	1759	1777	1795	1799	1802	1806
	1835	1863	1882	1900	1928	1942	1952	2024	2041	2100	2103	2160	2176	2214	2246
	2251	2258	2260	2283	2305	2318	2337	2340	2347	2350	2362	2363	2366	2379	2382
	2428	2432	2447	2453	2475	2483	2491	2496	2497	2498	2501	2506	2508	2530	2553
	2563	2573	2578	2580	2602	2606	2613	2615	2642	2652	2659	2675	2678	2685	2687
	2689	2704	2711	2720	2730	2754	2787	2797	2819	2827	2829	2838	2848	2867	2883
	2905	2939	2941	2961	2963	3032	3226	3239	3271	3290	3294	3312	3314	3341	3351
	3357	3371	3380	3407	3432	3439	3447	3460	3462	3479	3481	3487	3496	3504	3532
	3537	3580	3585	3628	3633	3676	3681	3784	3793	3797	3807	3826	3838	3849	3851
	3870	3872	3878	3880	3885	3888	3891	3892	4167	4168					
$IF	1#	1105	1147	1206	1218	1234	1251	1270	1293	1309	1326	1345	1367	1383	1400
	1419	1439	1464	1523	1554	1559	1597	1644	1661	1706	1723	1770	1787	1790	1853
	1870	1890	1931	1945	2035	2096	2170	2242	2312	2323	2326	2329	2351	2375	2422
	2462	2465	2468	2476	2484	2522	2549	2567	2598	2608	2646	2654	2661	2665	2669
	2680	2699	2706	2713	2722	2791	2811	2814	2821	2841	2907	2951	2956	3022	3273
	3302	3307	3343	3409	3434	3442	3468	3473	3490	3498	3522	3570	3618	3666	3787
	3801	3833													
$IFCOD	1#	1105	1147	1161	1206	1218	1234	1251	1270	1293	1309	1326	1345	1367	1383
	1400	1419	1439	1464	1523	1554	1559	1599	1644	1663	1706	1725	1770	1787	1790
	1853	1870	1890	1931	1945	2035	2098	2170	2244	2312	2323	2326	2329	2351	2353
	2377	2422	2451	2462	2465	2468	2476	2484	2522	2524	2551	2567	2600	2608	2646
	2654	2661	2665	2669	2680	2699	2706	2713	2722	2791	2811	2814	2821	2841	2871
	2891	2907	2951	2956	2965	3022	3230	3253	3273	3302	3307	3316	3343	3361	3373
	3384	3409	3434	3442	3458	3468	3473	3485	3490	3498	3522	3570	3618	3666	3787
	3801	3833													
$IFCON	1#	1502	1545	1628	1691	1753	1830	1923	2019	2155	2209	2276	2298	2637	2749
	2782	2934													
$IFOPR	1#	1106	1148	1162	1207	1219	1235	1252	1271	1294	1310	1327	1346	1368	1384
	1401	1420	1440	1465	1502	1524	1545	1555	1560	1600	1628	1645	1664	1691	1707
	1726	1753	1771	1788	1791	1830	1854	1871	1891	1923	1932	1946	2019	2036	2099
	2155	2171	2209	2245	2276	2298	2313	2324	2327	2330	2352	2354	2378	2423	2452
	2463	2466	2469	2477	2485	2523	2525	2552	2568	2601	2609	2637	2647	2655	2662
	2666	2670	2681	2700	2707	2714	2723	2749	2782	2792	2812	2815	2822	2842	2872

	2892	2908	2934	2952	2957	2966	3023	3231	3254	3274	3303	3308	3317	3344	3362
	3374	3385	3410	3435	3443	3459	3469	3474	3486	3491	3499	3523	3571	3619	3667
	3788	3802	3834												
$LET	1#	1079	1082	1085	1087	1090	1093	1096	1103	1112	1126	1129	1130	1131	1132
	1133	1135	1139	1142	1156	1158	1181	1182	1183	1184	1186	1187	1188	1189	1215
	1229	1231	1245	1248	1262	1265	1290	1304	1306	1320	1323	1337	1340	1364	1378
	1380	1394	1397	1411	1414	1436	1472	1477	1479	1487	1518	1527	1557	1565	1572
	1608	1612	1638	1672	1677	1704	1734	1739	1767	1811	1817	1839	1849	1868	1886
	1904	1911	1970	1974	1977	1980	1981	1982	1983	1984	1986	1999	2006	2045	2048
	2051	2055	2075	2076	2077	2078	2080	2081	2082	2083	2108	2109	2110	2111	2112
	2115	2117	2120	2127	2141	2186	2190	2193	2196	2223	2225	2226	2227	2228	2230
	2231	2232	2233	2287	2309	2403	2406	2408	2412	2414	2418	2420	2425	2430	2439
	2442	2458	2460	2528	2533	2546	2556	2558	2560	2565	2570	2576	2617	2620	2623
	2626	2644	2650	2657	2673	2683	2733	2736	2758	2768	2771	2825	2844	2846	2865
	2869	2879	2881	2885	2887	2889	2899	2901	2903	2911	2913	2916	2919	2922	2944
	2947	2949	2959	2979	2989	2992	2994	2997	3045	3048	3050	3053	3062	3065	3067
	3070	3078	3081	3083	3086	3094	3097	3099	3102	3110	3113	3115	3118	3126	3129
	3131	3134	3142	3145	3147	3150	3158	3161	3163	3166	3219	3224	3228	3235	3237
	3241	3243	3245	3247	3249	3251	3261	3264	3266	3269	3277	3280	3282	3285	3287
	3292	3296	3300	3310	3321	3323	3325	3327	3333	3336	3339	3346	3348	3353	3355
	3359	3366	3378	3382	3394	3396	3398	3400	3403	3405	3412	3415	3418	3421	3424
	3426	3429	3437	3445	3450	3453	3464	3466	3477	3516	3518	3520	3526	3529	3535
	3544	3546	3548	3550	3553	3557	3564	3566	3568	3574	3577	3583	3592	3594	3596
	3598	3601	3605	3612	3614	3616	3622	3625	3631	3640	3642	3644	3646	3649	3653
	3660	3662	3664	3670	3673	3679	3688	3690	3692	3694	3697	3701	3774	3776	3778
	3790	3795	3805	3823	3866	3902	3903	3904	3905	3906	3922	3923	3924	3925	3927
	3928	3929	3930	4052											
$LPCNT	1#	2256	3868	3876											
$OPADD	1#	1088	1091	1094	1103	1156	1159	1185	1557	1977	2079	2229	2259	2560	2565
	2825	2846	2889	2913	2944	2949	2959	3050	3067	3083	3099	3115	3131	3147	3163
	3245	3251	3277	3282	3287	3292	3300	3310	3359	3426	3445	3450	3453	3464	3477
	3518	3526	3546	3550	3566	3574	3594	3598	3614	3622	3642	3646	3662	3670	3690
	3694	3871	3879	3926	4052										
$OPAND	1#														
$OPCD1	1#	1088	1091	1094	1103	1156	1159	1185	1231	1248	1265	1306	1323	1340	1380
	1397	1414	1557	1704	1977	1999	2006	2048	2079	2120	2141	2190	2229	2259	2439
	2442	2458	2460	2533	2558	2560	2565	2570	2576	2623	2650	2657	2673	2683	2736
	2758	2825	2846	2889	2911	2913	2916	2922	2944	2947	2949	2959	3050	3053	3067
	3070	3083	3086	3099	3102	3115	3118	3131	3134	3147	3150	3163	3166	3245	3251
	3277	3280	3282	3285	3287	3292	3296	3300	3310	3359	3382	3415	3421	3426	3437
	3445	3450	3453	3464	3466	3477	3518	3526	3529	3535	3546	3550	3553	3557	3566
	3574	3577	3583	3594	3598	3601	3605	3614	3622	3625	3631	3642	3646	3649	3653
	3662	3670	3673	3679	3690	3694	3697	3701	3823	3871	3879	3926	4052		
$OPCD2	1#	2947	3296	3415	3421	3466									
$OPCOD	1#	1088	1091	1094	1103	1156	1159	1185	1231	1248	1265	1306	1323	1340	1380
	1397	1414	1557	1704	1977	1999	2006	2048	2079	2120	2141	2190	2229	2259	2439
	2442	2458	2460	2533	2558	2560	2565	2570	2576	2623	2650	2657	2673	2683	2736
	2758	2825	2846	2889	2911	2913	2916	2922	2944	2947	2949	2959	3050	3053	3067
	3070	3083	3086	3099	3102	3115	3118	3131	3134	3147	3150	3163	3166	3245	3251
	3277	3280	3282	3285	3287	3292	3296	3300	3310	3359	3382	3415	3421	3426	3437
	3445	3450	3453	3464	3466	3477	3518	3526	3529	3535	3546	3550	3553	3557	3566
	3574	3577	3583	3594	3598	3601	3605	3614	3622	3625	3631	3642	3646	3649	3653
	3662	3670	3673	3679	3690	3694	3697	3701	3823	3871	3879	3926	4052		
$OPCOM	1#														
$OPDEF	1#	1079	1082	1085	1087	1088	1090	1091	1093	1094	1096	1103	1105	1106	1109
	1112	1126	1129	1130	1131	1132	1133	1135	1139	1142	1147	1148	1156	1158	1159

	1161	1162	1181	1182	1183	1184	1185	1186	1187	1188	1189	1206	1207	1215	1218
	1219	1229	1231	1234	1235	1245	1248	1251	1252	1262	1265	1270	1271	1290	1293
	1294	1304	1306	1309	1310	1320	1323	1326	1327	1337	1340	1345	1346	1364	1367
	1368	1378	1380	1383	1384	1394	1397	1400	1401	1411	1414	1419	1420	1436	1439
	1440	1464	1465	1472	1477	1479	1487	1492	1493	1494	1495	1496	1497	1498	1502
	1518	1523	1524	1527	1531	1537	1538	1539	1540	1541	1542	1543	1545	1554	1555
	1557	1559	1560	1565	1569	1572	1576	1577	1579	1597	1598	1599	1600	1608	1612
	1618	1619	1620	1621	1622	1623	1624	1628	1638	1644	1645	1661	1662	1663	1664
	1672	1677	1682	1683	1684	1685	1686	1687	1688	1691	1704	1706	1707	1723	1724
	1725	1726	1734	1739	1744	1745	1746	1747	1748	1749	1750	1753	1767	1770	1771
	1787	1788	1790	1791	1794	1801	1811	1817	1820	1821	1822	1823	1824	1825	1826
	1830	1839	1842	1843	1844	1845	1849	1853	1854	1868	1870	1871	1886	1890	1891
	1904	1911	1913	1914	1915	1916	1917	1918	1919	1923	1931	1932	1945	1946	1970
	1974	1977	1980	1981	1982	1983	1984	1986	1989	1990	1991	1992	1993	1994	1995
	1999	2006	2009	2010	2011	2012	2013	2014	2015	2019	2028	2029	2030	2031	2035
	2036	2045	2048	2051	2055	2058	2059	2060	2061	2062	2063	2064	2075	2076	2077
	2078	2079	2080	2081	2082	2083	2096	2097	2098	2099	2108	2109	2110	2111	2112
	2115	2117	2120	2127	2132	2133	2134	2135	2136	2137	2138	2141	2145	2146	2147
	2148	2149	2150	2151	2155	2163	2164	2165	2166	2170	2171	2186	2190	2193	2196
	2200	2201	2202	2203	2204	2205	2206	2209	2223	2225	2226	2227	2228	2229	2230
	2231	2232	2233	2242	2243	2244	2245	2256	2257	2259	2261	2262	2268	2269	2270
	2271	2272	2273	2274	2276	2287	2290	2291	2292	2293	2294	2295	2296	2298	2309
	2312	2313	2323	2324	2326	2327	2329	2330	2339	2349	2351	2352	2353	2354	2365
	2375	2376	2377	2378	2403	2406	2408	2412	2414	2418	2420	2422	2423	2425	2427
	2430	2439	2442	2449	2450	2451	2452	2458	2460	2462	2463	2465	2466	2468	2469
	2474	2476	2477	2482	2484	2485	2490	2500	2522	2523	2524	2525	2528	2533	2546
	2549	2550	2551	2552	2556	2558	2560	2562	2565	2567	2568	2570	2572	2576	2598
	2599	2600	2601	2605	2608	2609	2617	2620	2623	2626	2629	2630	2631	2632	2633
	2634	2635	2637	2644	2646	2647	2650	2654	2655	2657	2661	2662	2665	2666	2669
	2670	2673	2677	2680	2681	2683	2699	2700	2706	2707	2713	2714	2722	2723	2733
	2736	2739	2740	2741	2742	2743	2744	2745	2749	2758	2760	2761	2762	2763	2768
	2771	2774	2775	2776	2777	2778	2779	2780	2782	2791	2792	2811	2812	2814	2815
	2818	2821	2822	2825	2841	2842	2844	2846	2865	2869	2871	2872	2879	2881	2885
	2887	2889	2891	2892	2899	2901	2903	2907	2908	2911	2913	2916	2919	2922	2925
	2926	2927	2928	2929	2930	2931	2934	2944	2947	2949	2951	2952	2954	2956	2957
	2959	2965	2966	2979	2989	2992	2994	2997	3022	3023	3045	3048	3050	3053	3062
	3065	3067	3070	3078	3081	3083	3086	3094	3097	3099	3102	3110	3113	3115	3118
	3126	3129	3131	3134	3142	3145	3147	3150	3158	3161	3163	3166	3219	3224	3228
	3230	3231	3235	3237	3241	3243	3245	3247	3249	3251	3253	3254	3261	3264	3266
	3269	3273	3274	3277	3280	3282	3285	3287	3289	3292	3296	3300	3302	3303	3305
	3307	3308	3310	3316	3317	3321	3323	3325	3327	3333	3336	3339	3343	3344	3346
	3348	3350	3353	3355	3359	3361	3362	3366	3373	3374	3378	3382	3384	3385	3394
	3396	3398	3400	3403	3405	3409	3410	3412	3415	3418	3421	3424	3426	3429	3434
	3435	3437	3442	3443	3445	3450	3453	3456	3457	3458	3459	3464	3466	3468	3469
	3471	3473	3474	3477	3483	3484	3485	3486	3490	3491	3498	3499	3516	3518	3520
	3522	3523	3526	3529	3531	3535	3544	3546	3548	3550	3553	3557	3564	3566	3568
	3570	3571	3574	3577	3579	3583	3592	3594	3596	3598	3601	3605	3612	3614	3616
	3618	3619	3622	3625	3627	3631	3640	3642	3644	3646	3649	3653	3660	3662	3664
	3666	3667	3670	3673	3675	3679	3688	3690	3692	3694	3697	3701	3774	3776	3778
	3787	3788	3790	3792	3795	3801	3802	3805	3810	3811	3812	3813	3818	3819	3820
	3821	3823	3825	3833	3834	3836	3840	3841	3850	3852	3866	3868	3869	3871	3873
	3874	3876	3877	3879	3881	3882	3884	3887	3893	3902	3903	3904	3905	3906	3922
	3923	3924	3925	3926	3927	3928	3929	3930	4052	4169					
$OPEQU	1#														
$OPNAN	1#														
$OPNEG	1#														

$OPNOR 1#
$OPNOT 1# 1248 1323 1397 1999 2048 2120 2190 2458 2460 2533 2558 2570 2758 3053
3070 3086 3102 3118 3134 3150 3166 3557 3605 3653 3701
$OPOR 1# 1231 1265 1306 1340 1380 1414 1704 2006 2141 2439 2442 2576 2623 2650
2657 2673 2683 2736 2911 2916 3280 3285 3437 3529 3577 3625 36~3
$OPROT 1#
$OPRO 1# 1079 1082 1085 1096 1112 1126 1129 1130 1131 1132 1133 1135 1139 1142
1181 1182 1183 1186 1187 1188 1189 1215 1229 1245 1262 1290 1304 1320 1337
1364 1378 1394 1411 1436 1472 1477 1479 1487 1518 1527 1565 1572 1608 1612
1638 1672 1677 1734 1739 1767 1811 1817 1839 1849 1868 1886 1904 1911 1970
1974 1980 1981 1982 1983 1984 1986 2045 2051 2055 2075 2076 2077 2080 2081
2082 2083 2108 2109 2110 2111 2112 2115 2117 2127 2186 2193 2196 2223 2225
2226 2227 2230 2231 2232 2233 2256 2287 2309 2403 2406 2403 2412 2414 2418
2420 2425 2430 2528 2546 2556 2617 2620 2626 2644 2733 2768 2771 2844 2865
2869 2879 2881 2885 2887 2899 2901 2903 2919 2979 2989 2992 2994 2997 3045
3048 3062 3065 3078 3081 3094 3097 3110 3113 3126 3129 3142 3145 3158 3161
3219 3224 3228 3235 3237 3241 3243 3247 3249 3261 3264 3266 3269 3321 3323
3325 3327 3333 3336 3339 3346 3348 3353 3355 3366 3378 3394 3396 3398 3400
3403 3405 3412 3418 3424 3429 3516 3520 3544 3548 3564 3568 3592 3596 3612
3616 3640 3644 3660 3664 3688 3692 3774 3776 3778 3790 3795 3805 3866 3868
3876 3902 3903 3904 3905 3906 3922 3923 3924 3927 3928 3929 3930
$OPR1 1# 2259 3871 3879
$OPR2 1# 1087 1090 1093 1103 1156 1158 1184 1231 1248 1265 1306 1323 1340 1380
1397 1414 1557 1704 1977 1999 2006 2048 2078 2120 2141 2190 2228 2439 2442
2458 2460 2533 2558 2560 2565 2570 2576 2623 2650 2657 2673 2683 2736 2758
2825 2846 2889 2911 2913 2916 2922 2944 2947 2949 2959 3050 3053 3067 3070
3083 3086 3099 3102 3115 3118 3131 3134 3147 3150 3163 3166 3245 3251 3277
3280 3282 3285 3287 3292 3296 3300 3310 3359 3382 3415 3421 3426 3437 3445
3450 3453 3464 3466 3477 3518 3526 3529 3535 3546 3550 3553 3557 3566 3574
3577 3583 3594 3598 3601 3605 3614 3622 3625 3631 3642 3646 3649 3653 3662
3670 3673 3679 3690 3694 3697 3701 3823 3925 4052
$OPSHF 1# 2947 3296 3415 3421 3466
$OPSUB 1# 2922 3382 3535 3553 3583 3601 3631 3649 3679 3697 3823
$OPSWB 1#
$OPXOR 1#
$OR 1# 1597 1661 1723 2096 2242 2375 2549 2598
$PUT 1# 1493 1538 1619 1683 1745 1821 1843 1914 1990 2010 2029 2059 2133 2146
2164 2201 2269 2291 2630 2740 2761 2775 2926 3819
$STRUC 1#
$SUBON 1# 1110 1114 1153 1161 1211 1224 1240 1257 1276 1299 1315 1332 1351 1373
1389 1406 1425 1450 1469 1508 1532 1550 1552 1567 1570 1574 1579 1580 1605
1633 1650 1669 1697 1713 1731 1759 1777 1795 1799 1802 1806 1835 1863 1882
1900 1928 1942 1952 2024 2041 2103 2160 2176 2214 2251 2260 2263 2283 2305
2318 2337 2340 2347 2350 2362 2363 2365 2366 2382 2428 2432 2449 2475 2483
2491 2496 2497 2498 2501 2506 2508 2530 2563 2573 2578 2580 2606 2613 2615
2642 2652 2659 2675 2678 2685 2687 2689 2704 2711 2720 2730 2754 2787 2797
2819 2827 2829 2838 2848 2871 2891 2939 2941 2961 2963 2965 3032 3230 3253
3290 3294 3312 3314 3316 3351 3357 3361 3373 3384 3439 3447 3456 3462 3479
3481 3483 3496 3504 3532 3537 3580 3585 3628 3633 3676 3681 3793 3797 3807
3825 3826 3838 3849 3872 3875 3880 3883 3884 3885 3887 3888 3891 4167
$THEN 1# 1105 1147 1206 1218 1234 1251 1270 1293 1309 1326 1345 1367 1383 1400
1419 1439 1464 1523 1554 1559 1599 1644 1663 1706 1725 1770 1787 1790 1853
1870 1890 1931 1945 2035 2098 2170 2244 2312 2323 2326 2329 2353 2377 2422
2462 2465 2468 2476 2484 2524 2551 2567 2600 2608 2646 2654 2661 2665 2669
2680 2699 2706 2713 2722 2791 2811 2814 2821 2841 2907 2951 2956 3022 3273
3302 3307 3343 3409 3434 3442 3468 3473 3490 3498 3522 3570 3618 3666 3787

```
        3801    3833
$TILA      1#
$TILO      1#
$UNTL2     1#   2449    3456    3483
$UNTL3     1#
$WHILE     1#
$$CMRE   885#
$$CMTM   885#
$$DEFA     1#
$$ENDS     1#
$$ERRO     1#
$$ESCA     1#    721#
$$GEN      1#   1110    1114    1137    1153    1211    1224    1240    1257    1276    1299    1315    1332    1351    1373
        1389    1406    1425    1450    1469    1474    1508    1532    1550    1552    1567    1570    1574    1580    1601
        1605    1633    1650    1665    1669    1697    1713    1727    1731    1759    1777    1795    1799    1802    1806
        1835    1863    1882    1900    1928    1942    1952    2024    2041    2100    2103    2160    2176    2214    2246
        2251    2258    2260    2283    2305    2318    2337    2340    2347    2350    2362    2363    2366    2379    2382
        2428    2432    2447    2453    2475    2483    2491    2496    2497    2498    2501    2506    2508    2530    2553
        2563    2573    2578    2580    2602    2606    2613    2615    2642    2652    2659    2675    2678    2685    2687
        2689    2704    2711    2720    2730    2754    2787    2797    2819    2827    2829    2838    2848    2867    2883
        2905    2939    2941    2961    2963    3032    3226    3239    3271    3290    3294    3312    3314    3341    3351
        3357    3371    3380    3407    3432    3439    3447    3460    3462    3479    3481    3487    3496    3504    3532
        3537    3580    3585    3628    3633    3676    3681    3772    3784    3793    3797    3807    3826    3838    3849
        3851    3863    3870    3872    3878    3880    3885    3888    3891    3892    4153    4167    4168
$$GETS     1#   1109    1110    1114    1153    1161    1211    1224    1240    1257    1276    1299    1315    1332    1351
        1373    1389    1406    1425    1450    1469    1508    1531    1532    1550    1552    1567    1569    1570    1574
        1576    1579    1580    1605    1633    1650    1669    1697    1713    1731    1759    1777    1794    1795    1799
        1801    1802    1806    1835    1863    1882    1900    1928    1942    1952    2024    2041    2103    2160    2176
        2214    2251    2260    2263    2283    2305    2318    2337    2339    2340    2347    2349    2350    2355    2362
        2363    2365    2366    2382    2427    2428    2432    2449    2474    2475    2478    2482    2483    2486    2490
        2491    2496    2497    2498    2500    2501    2506    2508    2530    2562    2563    2572    2573    2578    2580
        2605    2606    2613    2615    2642    2652    2659    2675    2677    2678    2685    2687    2689    2704    2711
        2720    2730    2754    2787    2797    2818    2819    2827    2829    2838    2848    2871    2891    2939    2941
        2961    2963    2965    3032    3230    3253    3289    3290    3294    3312    3314    3316    3350    3351    3357
        3361    3373    3384    3439    3447    3456    3462    3479    3481    3483    3496    3504    3531    3532    3537
        3579    3580    3585    3627    3628    3633    3675    3676    3681    3792    3793    3797    3807    3810    3825
        3826    3838    3849    3872    3875    3880    3883    3884    3885    3887    3888    3891    4167
$$GETT     1#   1109    1531    1569    1576    1794    1801    2339    2349    2355    2427    2474    2478    2482    2486
        2490    2500    2562    2572    2605    2677    2818    3289    3350    3531    3579    3627    3675    3792    3810
$$LPCN     1#   2259    3871    3879
$$NEWT     1#    721#   1115    1196    1279    1354    1428    1454    1587    1653    1715    1779    1957    2086    2235
        2368    2587    2805    2852    3202
$$POP      1#   1110    1114    1153    1161    1211    1224    1240    1257    1276    1299    1315    1332    1351    1373
        1389    1406    1425    1450    1469    1508    1532    1550    1552    1567    1570    1574    1579    1580    1605
        1633    1650    1669    1697    1713    1731    1759    1777    1795    1799    1802    1806    1835    1863    1882
        1900    1928    1942    1952    2024    2041    2103    2160    2176    2214    2251    2260    2263    2283    2305
        2318    2337    2340    2347    2350    2362    2363    2365    2366    2382    2428    2432    2449    2475    2483
        2491    2496    2497    2498    2501    2506    2508    2530    2563    2573    2578    2580    2606    2613    2615
        2642    2652    2659    2675    2678    2685    2687    2689    2704    2711    2720    2730    2754    2787    2797
        2819    2827    2829    2838    2848    2871    2891    2939    2941    2961    2963    2965    3032    3230    3253
        3290    3294    3312    3314    3316    3351    3357    3361    3373    3384    3439    3447    3456    3462    3479
        3481    3483    3496    3504    3532    3537    3580    3585    3628    3633    3676    3681    3793    3797    3807
        3825    3826    3838    3849    3872    3875    3880    3883    3884    3885    3887    3888    3891    4167
$$PUSH     1#   1105    1107    1111    1137    1138    1147    1149    1206    1208    1218    1220    1234    1236    1251
        1253    1270    1272    1293    1295    1309    1311    1326    1328    1345    1347    1367    1369    1383    1385
        1400    1402    1419    1421    1439    1441    1464    1466    1474    1475    1502    1503    1523    1525    1533
```

```
        1545    1546    1554    1556    1559    1561    1571    1579    1597    1602    1628    1629    1644    1646    1661
        1666    1691    1692    1706    1708    1723    1728    1753    1754    1770    1772    1787    1789    1790    1792
        1796    1803    1830    1831    1853    1855    1870    1872    1890    1892    1923    1924    1931    1933    1945
        1947    2019    2020    2035    2037    2096    2101    2155    2156    2170    2172    2209    2210    2242    2247
        2256    2258    2259    2263    2276    2277    2298    2299    2312    2314    2323    2325    2326    2328    2329
        2331    2341    2351    2355    2375    2380    2422    2424    2429    2447    2448    2462    2464    2465    2467
        2468    2470    2476    2478    2484    2486    2492    2502    2522    2526    2549    2554    2564    2567    2569
        2574    2598    2603    2607    2608    2610    2637    2638    2646    2648    2654    2656    2661    2663    2665
        2667    2669    2671    2679    2680    2682    2699    2701    2706    2708    2713    2715    2722    2724    2749
        2750    2782    2783    2791    2793    2811    2813    2814    2816    2820    2821    2823    2841    2843    2867
        2868    2883    2884    2905    2906    2907    2909    2934    2935    2951    2953    2956    2958    3022    3024
        3226    3227    3239    3240    3271    3272    3273    3275    3291    3302    3304    3307    3309    3341    3342
        3343    3345    3352    3371    3372    3380    3381    3407    3408    3409    3411    3432    3433    3434    3436
        3442    3444    3468    3470    3473    3475    3490    3492    3498    3500    3522    3524    3533    3570    3572
        3581    3618    3620    3629    3666    3668    3677    3773    3784    3785    3787    3789    3794    3801    3803
        3825    3833    3835    3864    3868    3870    3871    3875    3876    3878    3879    3883    4154
$$SELE     1#
$$SET   4770#   4779    4780    4781    4782    4784    4786    4787    4788
$$SETM  1039#
$$SETS     1#   1105    1107    1111    1137    1138    1147    1149    1206    1208    1218    1220    1234    1236    1251
        1253    1270    1272    1293    1295    1309    1311    1326    1328    1345    1347    1367    1369    1383    1385
        1400    1402    1419    1421    1439    1441    1464    1466    1474    1475    1502    1503    1523    1525    1533
        1545    1546    1554    1556    1559    1561    1571    1579    1597    1602    1628    1629    1644    1646    1661
        1666    1691    1692    1706    1708    1723    1728    1753    1754    1770    1772    1787    1789    1790    1792
        1796    1803    1830    1831    1853    1855    1870    1872    1890    1892    1923    1924    1931    1933    1945
        1947    2019    2020    2035    2037    2096    2101    2155    2156    2170    2172    2209    2210    2242    2247
        2256    2258    2259    2263    2276    2277    2298    2299    2312    2314    2323    2325    2326    2328    2329
        2331    2341    2351    2355    2375    2380    2422    2424    2429    2447    2448    2462    2464    2465    2467
        2468    2470    2476    2478    2484    2486    2492    2502    2522    2526    2549    2554    2564    2567    2569
        2574    2598    2603    2607    2608    2610    2637    2638    2646    2648    2654    2656    2661    2663    2665
        2667    2669    2671    2679    2680    2682    2699    2701    2706    2708    2713    2715    2722    2724    2749
        2750    2782    2783    2791    2793    2811    2813    2814    2816    2820    2821    2823    2841    2843    2867
        2868    2883    2884    2905    2906    2907    2909    2934    2935    2951    2953    2956    2958    3022    3024
        3226    3227    3239    3240    3271    3272    3273    3275    3291    3302    3304    3307    3309    3341    3342
        3343    3345    3352    3371    3372    3380    3381    3407    3408    3409    3411    3432    3433    3434    3436
        3442    3444    3468    3470    3473    3475    3490    3492    3498    3500    3522    3524    3533    3570    3572
        3581    3618    3620    3629    3666    3668    3677    3773    3784    3785    3787    3789    3794    3801    3803
        3825    3833    3835    3864    3868    3870    3871    3875    3876    3878    3879    3883    4154
$$SETT     1#   2355    2478    2486
$$SKIP     1#    721#   1209    1467    1582    1603    1667    1729    1797    1804    1861    1880    1898    1940    1955
        2249    2281    2303    2335    2345    2360    2510    2802    3035
.EQUAT     1#    588#    611
.HEADE     1#    588#    589
.KT11      1#
.SETUP     1#    588#   1002
.SWRHI     1#    588#    600
.SWRLO   611#
.$ACT1     1#    588#    852
.$APTB     1#    588#    929#
.$APTH     1#    588#    863
.$APTY     1#    588#   4428
.$ASTA     1#
.$CATC     1#    588#    825
.$CMTA     1#    588#    885
.$DB2D     1#
.$DB2C     1#
```

```
.$DIV       1#
.$EOP       1#     588#    4171
.$ERRO      1#     588#    4485
.$ERRT      1#
.$MULT      1#
.$POWE      1#
.$RAND      1#
.$RDDE      1#
.$RDOC      1#
.$READ      1#     588#    4289
.$R2AZ      1#
.$SAVE      1#
.$SB2D      1#
.$SB2O      1#
.$SCOP      1#     588#    4539
.$SIZE      1#
.$SUPR      1#
.$TRAP      1#     588#    4747
.$TYPB      1#
.$TYPD      1#     588#    4603
.$TYPE      1#     588#    4210
.$TYPO      1#     588#    4670
.$40CA      1#
.1170       1#


. ABS.  017354     000


 ERRORS DETECTED:  0

CVDLAA,CVDLAA=SPMAC.SML,SYSMAC.SML,CVDLAA.P11
RUN-TIME: 105 112 7 SECONDS
RUN-TIME RATIO: 996/226=4.4
CORE USED:  42K  (83 PAGES)
```